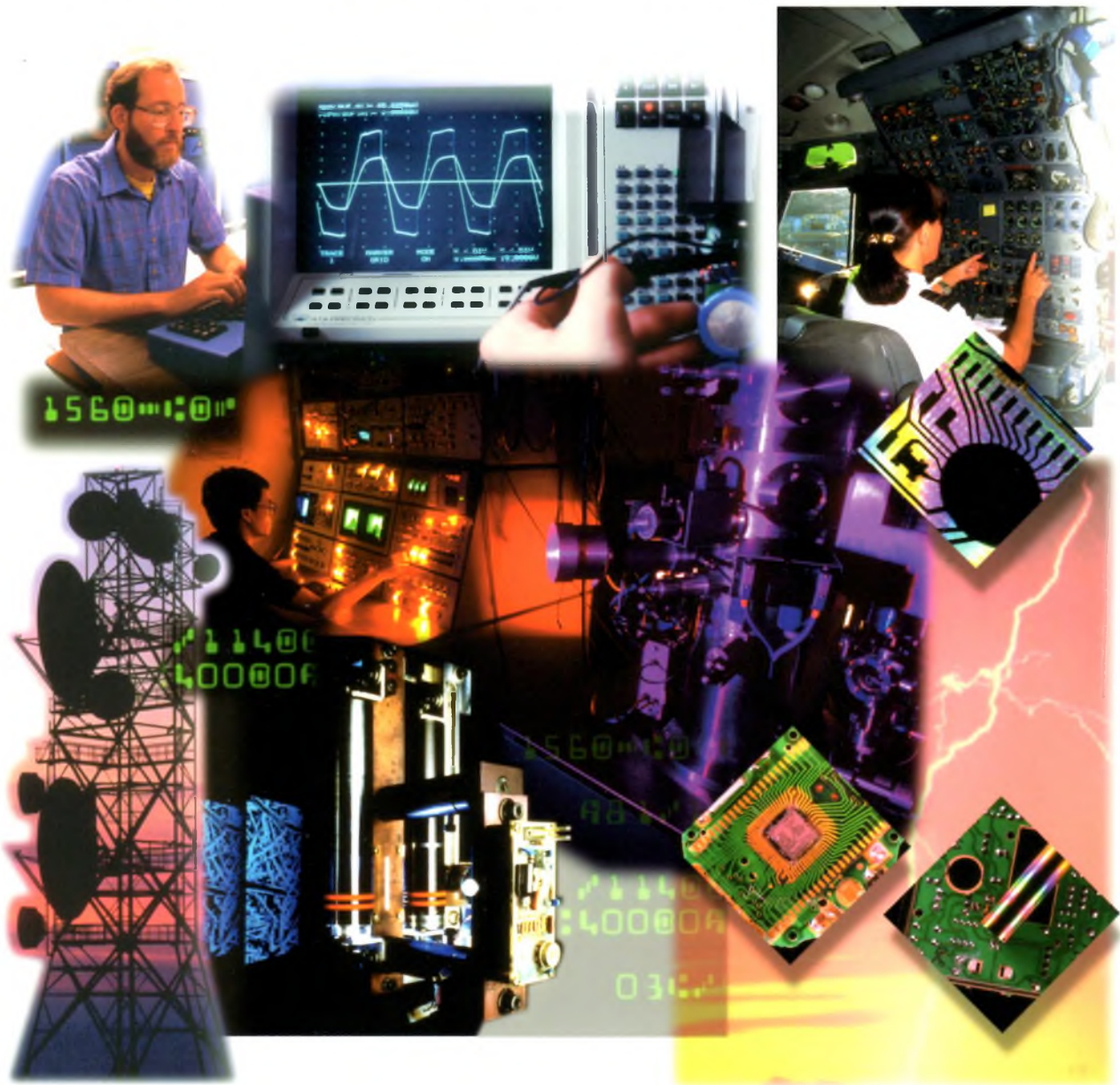


INDIVIDUAL LEARNING SYSTEM



T
E
X
T
B
O
O
K

DIGITAL TECHNIQUES



IMPORTANT NOTICE

The 74LS95 shift register IC used in experiments 15, 16, and 22 may operate erratically without a slight circuit modification. Therefore, in each experiment you need to wire-in two 200 pF bypass capacitors, as shown in Figure 1. One capacitor should be wired between pin 6 and ground, and the other capacitor between pin 14 and ground.

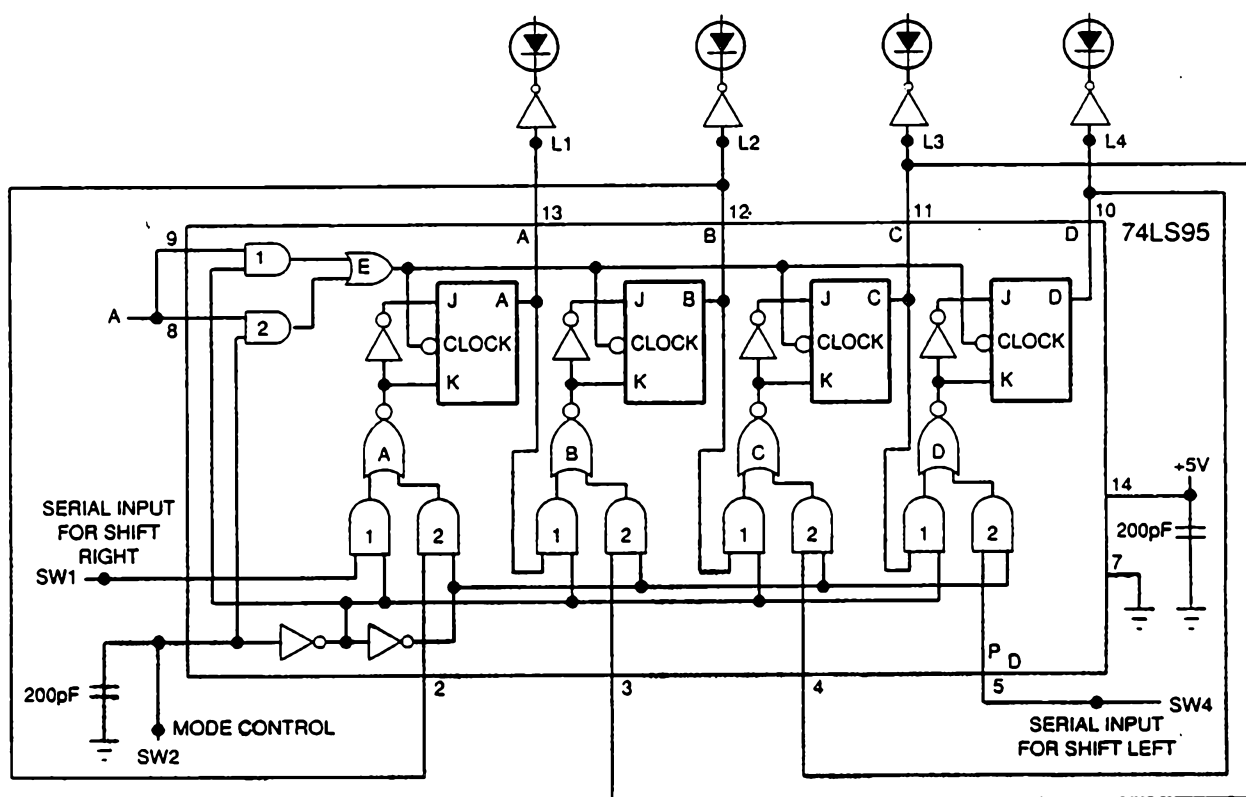


Figure 1
Example of where to wire bypass capacitors on a 74LS95 IC.

Thank you,
HEATHKIT COMPANY

DIGITAL TECHNIQUES

Model EE-3201A
HEATHKIT COMPANY, INC.
BENTON HARBOR, MICHIGAN 49022
595-2931-09

Copyright © 1998, 1993, 1983
Heathkit Company, Inc.
All Rights Reserved
Printed in the United States of America

ISBN 0-87119-011-7

CONTENTS

COURSE INTRODUCTION	III
COURSE OBJECTIVES	VI
COURSE OUTLINE	VIII
PARTS LIST	XVIII
UNIT ONE — Introduction to Digital Techniques	1-1
UNIT TWO — Semiconductor Devices for Digital Circuits	2-1
UNIT THREE — Digital Logic Circuits	3-1
UNIT FOUR — Digital Integrated Circuits	4-1
UNIT FIVE — Boolean Algebra	5-1
UNIT SIX — Flip-Flops and Registers	6-1
UNIT SEVEN — Sequential Logic Circuits: Counters, Shift Registers and Clocks	7-1
UNIT EIGHT — Combinational Logic Circuits	8-1
UNIT NINE — Semiconductor Memories	9-1
UNIT TEN — Data Conversion	10-1
UNIT ELEVEN — Digital Troubleshooting	11-1

INTRODUCTION

This is the computer age and if you are to succeed or excel in electronics, you must have a knowledge of digital techniques. All computers and most other electronic circuits use digital techniques. The application of the material in this course will be beneficial, if not essential to your understanding of digital equipment. This course will teach you the concepts, terminology, components, and circuits that combine to form the basic digital system.

Digital techniques are so widely used today that it is almost impossible to think of electronic equipment without them. Digital techniques are used in virtually every area of electronics. They have greatly improved electronic methods and have resulted in practical electronic equipment with amazing capability. And, there is potential for further improvements and advances. As an electronic engineer, technician, or hobbyist, you can benefit by knowing digital techniques. This program will provide you with a solid understanding of digital methods and a guide to their application.

After completing this course, you will be familiar with a wide range of integrated circuits, their uses and characteristics. You will also have a working knowledge of semiconductor devices, Boolean algebra, logic circuits, memory devices, data conversion, and digital troubleshooting.

How do you gauge your learning? Let the “**objectives**” be your guide. These carefully constructed objectives are the framework for the course. When you can meet all of the objectives, you have satisfied the requirements of the course. You’ll find two types of objectives in this course: broad, “**Course Objectives**” are listed following this introduction. More specific “**Unit Objectives**” are listed near the front of each unit. When you can satisfy these unit objectives, you’ve learned everything that was intended from the units; no matter how easy it seemed.

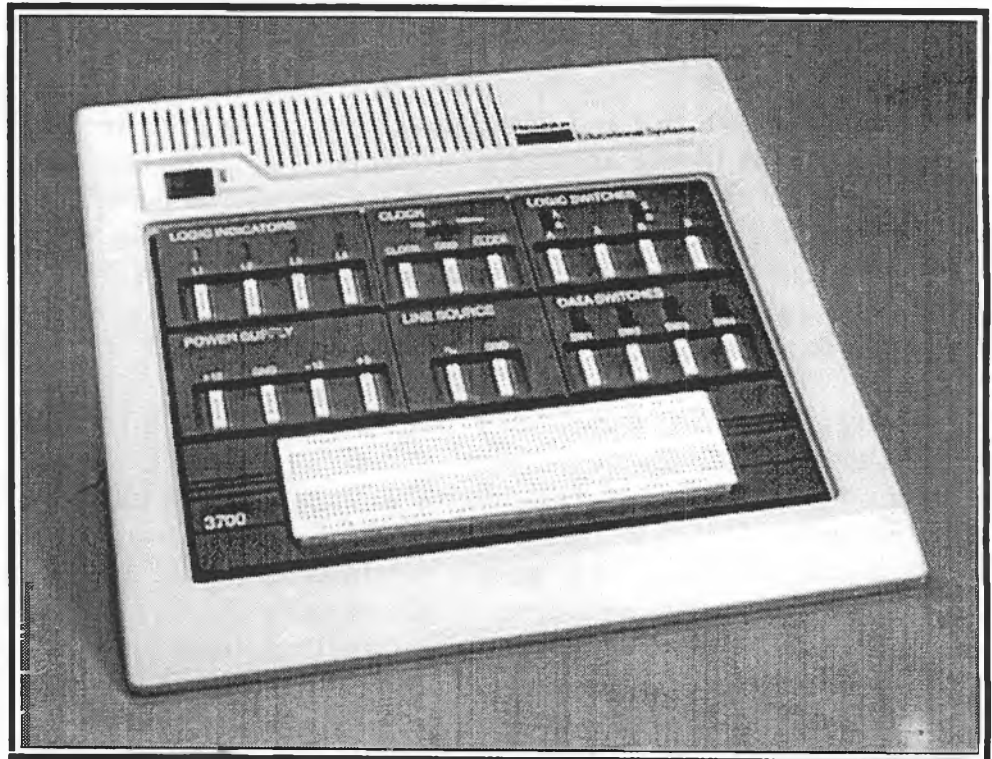


Figure 1
The ET-3700 Electronic Digital Design Experimenter.

To perform the Experiments in this program, you will need either the ET-3200 or the ET-3700 Electronic Design Experimenter. The ET-3700 is shown in Figure 1. In addition, you will also need a volt-ohm meter and an oscilloscope similar to those shown in Figure 2. All other parts, such as ICs, resistors, and capacitors, have been supplied in the Parts Pack. Be sure to check your parts against the "Parts List" that follows the "Course Outline." If you are missing any parts, you must request them on the Parts Order Form provided.

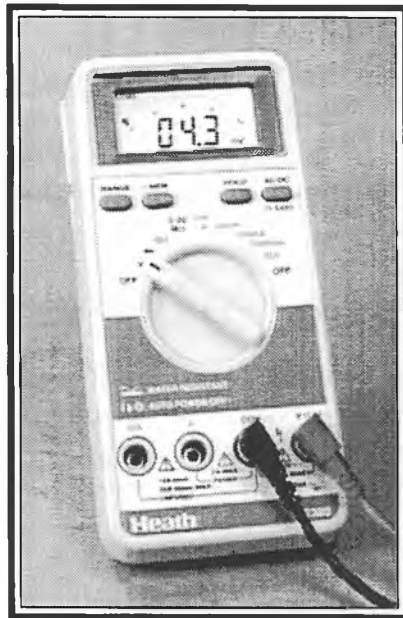
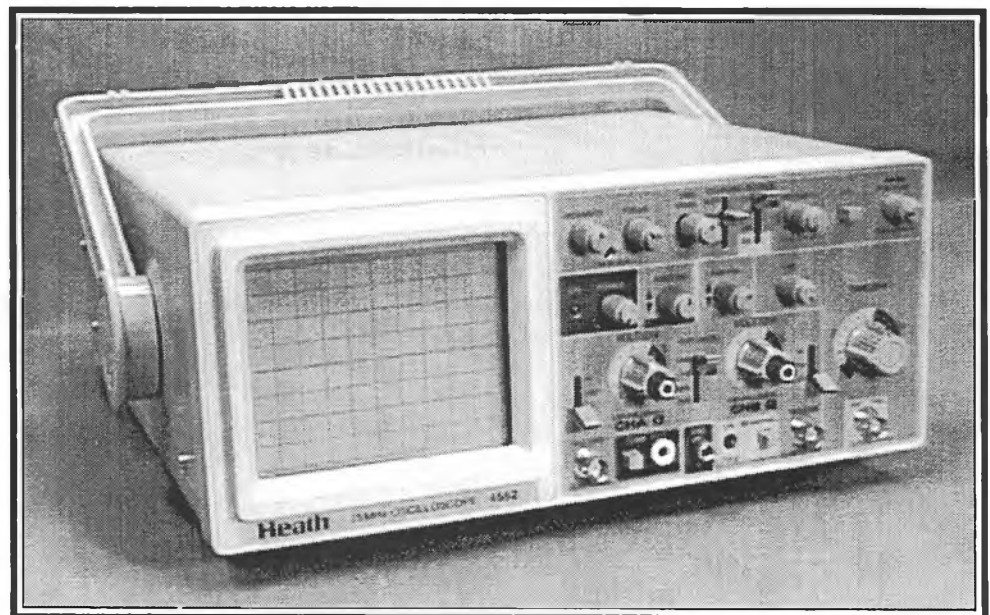


Figure 2
Typical volt-ohm meter (multimeter)
and oscilloscope.



COURSE OBJECTIVES

When you complete this program, you will have the following skills and knowledge. You will be able to:

1. Discuss the advantages and benefits of using digital techniques in electronic equipment.
2. Name the major applications of digital techniques in electronics.
3. Convert between the binary and decimal number systems and recognize the most commonly used binary codes.
4. Name the major components used in implementing digital circuits and explain how they operate.
5. Explain the operation of digital logic gates.
6. Identify the more commonly used integrated circuit families used in digital equipment and discuss their operation, characteristics, and features.
7. Use Boolean algebra to express logic operations and minimize logic circuits in design.
8. Explain the operation of flip-flops.
9. Discuss the operation and application of binary and BCD counters, shift registers, and other sequential logic circuits.

10. Name the most frequently used combinational logic circuits and explain their operation.
11. Design both combinational and sequential logic circuits for a given application from definition and concept to the selection of the integrated circuits.
12. Identify the various types of semiconductor memories (RAMs, ROMs, bubbles, etc.), explain how they operate and give examples as to how they are used.
13. Name the various types of data conversion such as digital-to-analog converters, analog-to-digital converters, multiplexers, and sample/hold circuits, tell how they operate and give examples of their application.
14. Troubleshooting digital circuits using standard test equipment and special instruments such as logic probes, logic and signature analyzers.

COURSE OUTLINE

UNIT 1 INTRODUCTION TO DIGITAL TECHNIQUES

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Digital Techniques
 - A. Contrasting Analog and Digital Devices and Techniques
 - B. Where are Digital Techniques Used?
 - 1. Communications
 - 2. Telemetry Systems
 - 3. Test Instruments
 - 4. Industrial Controls
 - 5. Consumer Electronic Equipment
 - C. Why Use Digital Techniques?
 - 1. Greater Accuracy
 - 2. Greater Dynamic Range
 - 3. Greater Stability
 - 4. Convenience
 - 5. Automation
 - 6. Design Simplicity
 - 7. New Approaches
- V. The Binary Number System
 - A. Positional Number Systems
 - B. Fractional Numbers
 - C. Converting Between the Binary and Decimal Number Systems
 - 1. Binary to Decimal
 - 2. Decimal to Binary
 - E. Binary Number Sizes
 - F. Number Identification
- VI. Binary Codes
 - A. Binary Coded Decimal
 - B. Special Binary Codes
 - 1. Gray Code
 - 2. ASCII Code

- VII. Data Representation
 - A. Electromechanical Devices
 - B. Transistors
 - C. Logic Levels
 - D. Positive and Negative Logic
 - E. Parallel vs Serial Data Representation
 - F. Logic Circuits
- VIII. Unit Examination
- IX. Examination Answers

UNIT 2 SEMICONDUCTOR DEVICES FOR DIGITAL CIRCUITS

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. The Bipolar Transistor Switch
 - A. Modes of Operation
 - 1. Cut-off
 - 2. Linear
 - 3. Saturation
 - B. Saturated Switching Circuits
 - C. Switching Speed
 - D. Non-Saturating Switching Circuits
- V. Designing a Saturated Switch Logic Inverter
 - A. Procedure
 - B. Example Application 1
 - C. Example Application 2
- VI. Experiment 1 Bipolar Transistor Switch
- VII. MOS Field Effect Transistors
 - A. The N-Channel MOSFET
 - B. The P-Channel MOSFET
 - C. Bipolars vs MOSFET's
 - D. MOSFET Circuits
- VIII. Unit Examination
- IX. Examination Answers

UNIT 3 DIGITAL LOGIC CIRCUITS

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Types of Logic Circuits
- V. The Inverter
- VI. Experiment 2 Logic Inverter
- VII. Decision-Making Logic Elements
 - A. The AND Gate
 - B. The OR Gate
 - C. The Dual Nature of Logic Gates
- VIII. Experiment 3 Diode Logic Gates
- IX. NAND/NOR Gates
 - A. NAND Gate
 - B. NOR Gate
 - C. How NAND/NOR Gates Are Used
- X. Practical Logic Circuits
 - A. Relays and Switches
 - B. Discrete Component Logic Circuits
 - C. Integrated Circuits
- XI. Experiment 4 Transistor Logic Gate
- XII. Unit Examination
- XIII. Examination Answers

UNIT 4 DIGITAL INTEGRATED CIRCUITS

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Logic Circuit Characteristics
 - A. Logic Levels
 - B. Propagation Delay
 - C. Power Dissipation
 - D. The Speed-Power Trade-Off
 - E. Noise Immunity
 - F. Fan Out
 - 1. Current Source Logic
 - 2. Current Sink Logic

- V. Integrated Circuits**
 - A. Manufacturing Methods**
 - 1. Monolithic
 - 2. Thin and Thick Film Techniques
 - 3. Hybrid Circuits
 - B. Application**
 - C. Function**
 - D. Integrated Circuit Packaging**
 - 1. TO5
 - 2. Flat-Pack
 - 3. DIP
 - E. Temperature Ranges**
- VI. Transistor Transistor Logic**
 - A. Circuit Operation**
 - B. TTL Characteristics**
 - C. Special TTL Variations**
 - 1. Low Power TTL
 - 2. High Power TTL
 - 3. Schottky TTL
 - 4. Three State TTL and Data Busses
- VII. Experiment 5 TTL Logic Gates**
- VIII. Emitter Coupled Logic**
 - A. Circuit Operation**
 - B. ECL Characteristics**
- IX. Metal Oxide Semiconductor Integrated Circuits**
 - A. PMOS and NMOS Circuits**
 - B. Complementary MOS**
 - C. CMOS Characteristics**
- X. Experiment 6 CMOS Logic Gate**
- XI. Integrated Injection Logic**
- XII. Selecting a Digital Integrated Circuit for a Specific Application**
 - A. Trends**
 - B. Complex Functions**
 - C. Trade-Offs**
- XIII. Unit Examination**
- XIV. Examination Answers**

UNIT 5 BOOLEAN ALGEBRA

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Relating Digital Logic Circuits and Boolean Equations
 - A. Review of Basic Functions
 - B. Boolean Formats
 - C. Sum-of-Products
 - D. Product-of-Sums
- V. Truth Tables
- VI. Boolean Rules
- VII. Minimizing Logic Expressions
- VIII. Using NAND/NOR Gates
- IX. Experiment 7 Applying NAND and NOR Gates
- X. Experiment 8 The Wired OR Connection
- XI. Unit Examination
- XII. Examination Answers

UNIT 6 FLIP-FLOPS AND REGISTERS

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Flip-Flops
- V. Experiment 9 Set-Reset Flip-Flops
- VI. D Type Flip-Flops
- VII. Storage Registers
- VIII. Experiment 10 D Type Flip-Flops and Registers
- IX. JK Flip-Flops
- X. Experiment 11 JK Flip-Flops
- XI. Unit Examination
- XII. Examination Answers

UNIT 7 SEQUENTIAL LOGIC CIRCUITS: COUNTERS, SHIFT REGISTERS, AND CLOCKS

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide

- IV. Counters**
 - A. Binary Counters**
 - 1. Frequency Dividers
 - 2. Maximum Count
 - 3. Down Counters
 - 4. Up-Down Counter
 - 5. Synchronous Counters
 - 6. Counter Control Functions
 - B. Typical Integrated Circuit Counters**
- V. Experiment 12 Binary Counters**
 - A. BCD Counters**
 - 1. Cascading BCD Counters
 - 2. The BCD Counter as a Frequency Divider
 - 3. Typical Integrated Circuit BCD Counter
- VI. Experiment 13 The BCD Counter**
 - A. Special Counters**
 - 1. Modulo 3 Counter
 - 2. Modulo 5 Counter
- VII. Experiment 14 Counter Applications**
- VIII. Shift Registers**
 - A. Shift Register Operation**
 - B. Bipolar Logic Shift Registers**
- IX. Experiment 15 Shift Registers**
 - A. Shift Register Applications**
 - 1. Scaling Operations
 - 2. Shift Register Memory
 - 3. Sequencer/Ring Counter
 - 4. Counters
- X. Experiment 16 Shift Register Applications**
 - A. MOS Shift Registers**
 - 1. Dynamic MOS Shift Registers
 - 2. Static MOS Shift Registers
- XI. Clocks and One Shots**
 - A. Clock Oscillator Circuits**
 - 1. Discrete Component Circuits
 - 2. IC Clock Circuits
 - B. One Shot Multivibrators**
 - C. One Shot Applications**
- XII. Experiment 17 Clocks and One Shots**
- XIII. Unit Examination**
- XIV. Examination Answers**

UNIT 8 COMBINATIONAL LOGIC CIRCUITS

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Decoders
 - A. BCD to Decimal Decoder
 - B. Octal and Hex Decoders
 - C. BCD to 7-Segment Decoder
- V. Experiment 18 Decoders
- VI. Encoders
- VII. Experiment 19 7-Segment Decoder-Driver and Display
- VIII. Multiplexers
 - A. Multiplexer Operation
 - B. Multiplexer Applications
 - 1. Parallel to Serial Conversion
 - 2. Serial Binary Word Generator
 - 3. Boolean Function Generator
- IX. Experiment 20 Multiplexers
- X. Demultiplexers
- XI. Exclusive OR
 - A. Exclusive NOR
 - B. Applications of the Exclusive OR
 - 1. Binary Adder
 - 2. Parity Generator/Checker
 - 3. Binary Comparators
- XII. Experiment 21 Exclusive OR
- XIII. Code Converters
- XIV. Experiment 22 Exclusive OR Applications
- XV. Read Only Memories
 - A. ROM Operation
 - B. ROM construction
 - 1. Bipolar ROM
 - 2. MOS ROM's
 - 3. Access Time
 - C. ROM Applications
 - 1. Random Logic
 - 2. Code Conversion
 - 3. Arithmetic Operations
 - D. Microprogramming
- XVI. Programmable Logic Arrays
- XVII. Unit Examination
- XVIII. Examination Answers

UNIT 9 SEMICONDUCTOR MEMORIES

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Memory Types and Organization
 - A. Memory Classification
 - B. Memory Organization
- V. Memory Characteristics and Specifications
 - A. Volatility
 - B. Access Time
 - C. Memory Size
 - D. Memory Configuration
 - E. Device Technology
- VI. Read/Write Memories
 - A. Static Memories
 - B. RAM Organization
 - C. R/W Memories
 - D. Typical Memory Configurations
- VII. Dynamic Memories
 - A. Dynamic RAM Organization
 - B. 64k and Larger RAMs
- VIII. Programmable Read Only Memories
 - A. Bipolar PROMs
 - B. MOS PROMs
 - C. PROM Programmers
- IX. Experiment 23 Semiconductor Memories
- X. Unit Examination
- XI. Examination Answers

UNIT 10 DATA CONVERSION

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Purpose of Data Conversion
- V. Digital-to-Analog Conversion
 - A. Binary Weighted Resistor DAC
 - B. Improved Weighted Resistor DAC
 - C. R/2R Ladder DAC
 - D. DAC Specifications and Error Sources
 - E. Multiplying DACs
- VI. Experiment 24 Digital-to-Analog Conversion
- VII. Analog-to-Digital Conversion
 - A. Comparators
 - B. Counter-Ramp Feedback ADC
 - C. Successive Approximation Converter
 - D. Flash Converter
 - E. Dual Slope ADC
 - F. Voltage-to-Frequency Conversion
 - G. ADC Specifications
 - H. Sampling and Multiplexing
- VIII. Experiment 25 Analog-to-Digital Conversion
- IX. Unit Examination
- X. Examination Answers

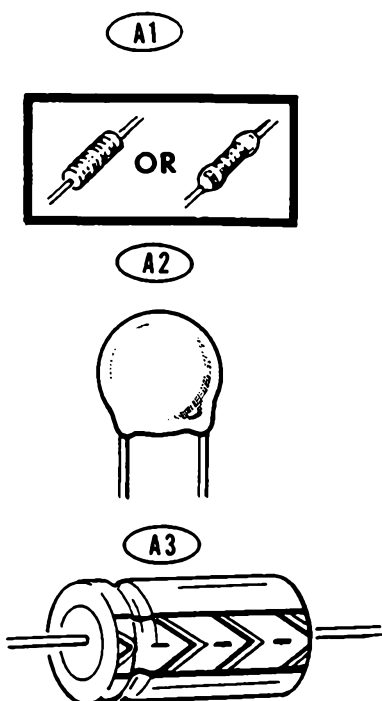
UNIT 11 DIGITAL TROUBLESHOOTING

- I. Introduction
- II. Unit Objectives
- III. Unit Activity Guide
- IV. Typical Problems in Digital Circuits
 - A. Operator Problems
 - B. Construction Problems
 - C. Defective Components
 - D. Mechanical Problems
 - E. Power Supply Problems
 - F. Timing Problems
 - G. Environmental Problems
- V. Digital IC Problems
 - A. External IC Problems
 - B. Internal IC Problems
- VI. Digital Test Instruments
 - A. Multimeters
 - B. Oscilloscopes
 - C. Logic Clips and Monitors
 - D. Logic Probe
 - E. Logic Pulser
 - F. Current Tracer
 - G. Logic Analyzer
 - H. Signature Analyzer
- VII. Procedures for Digital Troubleshooting
 - A. Data Collection
 - B. Isolating the Problem
 - C. Making the Repair
 - D. Final Testing
- VIII. Experiment 26 Practical Digital Troubleshooting
- IX. Unit Examination
- X. Examination Answers

PARTS LIST

This parts list contains all of the parts used in experiments which you will perform with this course. The key number in the parts list corresponds to the numbers in the parts pictorial. Some parts are packaged in envelopes. Except for this initial parts check, keep these parts in their envelopes until they are called for in the experiment. A container is provided so that you can keep the small parts together in one place.

KEY No.	PART No.	QTY.	DESCRIPTION
------------	-------------	------	-------------



RESISTORS (5%)

A1	6-102-12	4	1000 Ω (brown-black-red)
A1	6-472-12	4	4700 Ω (yellow-violet-red)
A1	6-103-12	2	10 k Ω (brown-black-orange)
A1	6-473-12	2	47 k Ω (yellow-violet-orange)
A1	6-224-12	1	220 k Ω (red-red-yellow)
A1	6-151-12	1	150 Ω (brown-green-brown)
A1	6-561-12	1	560 Ω (green-blue-brown)
A1	6-153-12	1	15 k Ω (brown-green-orange)
	10-311	1	5 k Ω Control
	10-312	2	10 k Ω Control

CAPACITORS

Disc

A2	21-21	2	200 pF
A2	21-192	1	0.1 μ F

Electrolytic

A3	25-875	2	1000 μ F
----	--------	---	--------------

KEY No.	PART No.	QTY.	DESCRIPTION
------------	-------------	------	-------------

DIODES-LED DISPLAY TRANSISTORS

A4	56-56	4	1N4149 silicon diode
A4	56-59	1	4.7 V Zener
A5	411-885	1	7 segment LED display
A6	417-801	2	MPSA20 transistor

NOTE: Transistors and integrated circuits are marked for identification in one of the following ways:

1. Part number.
2. Type number.
3. Part number and type number.
4. Part number with a type number other than the one listed.

INTEGRATED CIRCUITS

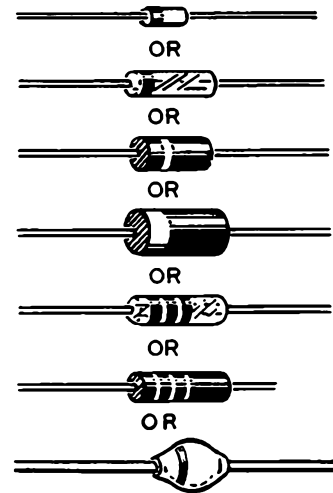
A7	442-21	1	1458
A7	442-751	1	1408
A7	443-1802	1	14495-1
A7	443-695	1	4001
A7	443-728	2	74LS00
A7	443-745	1	74LS03
A7	443-755	1	74LS04
A7	443-764	1	2114
A7	443-779	1	74LS02
A7	443-780	1	74LS08
A7	443-781	1	74LS75
A7	443-798	1	74LS20
A7	443-807	1	74LS42
A7	443-813	1	74LS90
A7	443-814	1	74LS95
A7	443-942	1	74LS123
A7	443-815	2	74LS193
A7	443-829	2	74LS76
A7	443-878	1	74LS151
A7	443-891	1	74LS86

MISCELLANEOUS

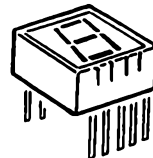
266-692	1	Small parts container
344-52	10 Ft.	Red wire
490-111	1	IC puller

A4

NOTE: HEATH PART NUMBERS ARE STAMPED ON MOST DIODES.



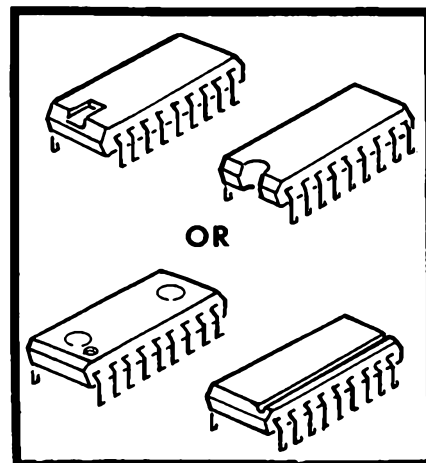
A5



A6



A7



Unit 1

**INTRODUCTION TO
DIGITAL TECHNIQUES**

CONTENTS

Introduction	1-3
Unit Objectives	1-4
Unit Activity Guide	1-5
Digital Techniques	1-6
The Binary Number System	1-21
Binary Codes	1-31
Data Representation	1-40
Unit Examination	1-47
Examination Answers	1-51

INTRODUCTION

The purpose of this first unit on digital techniques is to give you an overview of the subject and to introduce you to the basic concepts. You will learn what digital techniques are, how they are used and why they are used. You will learn about binary numbers and codes which are the basic language of all digital systems. And finally, you will see how digital techniques are implemented with hardware.

Digital techniques are so widely used today that it is almost impossible to think of electronic equipment without them. Digital techniques are used in virtually every area of electronics. They have greatly improved electronic methods and have given us practical electronic equipment with amazing capability. And, there is potential for further improvements and advances. As an electronic engineer, technician, or hobbyist you can benefit by knowing digital techniques. This program will provide you with a solid understanding of digital methods and a guide to their application.

Examine the Unit Objectives listed in the next section to see what you will learn in this unit. Then follow the instruction in the Unit Activity Guide to be sure you perform all of the steps necessary to complete this lesson successfully. Check off each step as you complete it. In the spaces provided, keep track of the time you spend on each activity.

UNIT OBJECTIVES

When you complete this Unit you will have the following knowledge and capabilities:

1. Given a list of physical variables, components, devices, and other items, you will be able to classify them as being either analog or digital in nature.
2. You will be able to list at least five advantages of digital techniques over analog methods.
3. You will be able to list at least five examples of electronic equipment using digital techniques.
4. You will be able to state the factors that have most influenced the growth of digital techniques.
5. Given any decimal number, you will be able to convert it into its binary equivalent.
6. Given any binary number, you will be able to convert it into its decimal equivalent.
7. Given any decimal number, you will be able to convert it into its binary coded decimal (BCD) equivalent.
8. Given a BCD number you will be able to convert it into its decimal equivalent.
9. You will be able to list four popular digital codes.
10. You will be able to list the two key ways binary data is represented with digital hardware.
11. You will be able to list the advantages and disadvantages of both serial and parallel methods of binary data transmission.
12. You will be able to identify binary signals as being either positive or negative logic.

UNIT ACTIVITY GUIDE

	Completion Time
<input type="checkbox"/> Read "Digital Techniques."	_____
<input type="checkbox"/> Answer Self Test Review Questions 1-8.	_____
<input type="checkbox"/> Read "The Binary Number System."	_____
<input type="checkbox"/> Answer Self Test Review Questions 9-15.	_____
<input type="checkbox"/> Read "Binary Codes."	_____
<input type="checkbox"/> Answer Self Test Review Questions 16-22.	_____
<input type="checkbox"/> Read "Data Representation."	_____
<input type="checkbox"/> Answer Self Test Review Questions 23-27.	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Review the Examination Answers.	_____

DIGITAL TECHNIQUES

There are two basic types of electronic signals and techniques, analog and digital. Analog signals are the most familiar type. An analog signal is an ac or dc voltage or current that varies smoothly or continuously. It is one that does not change abruptly or in steps. An analog signal can exist in a wide variety of forms. Several types of analog signals are shown in Figure 1-1.

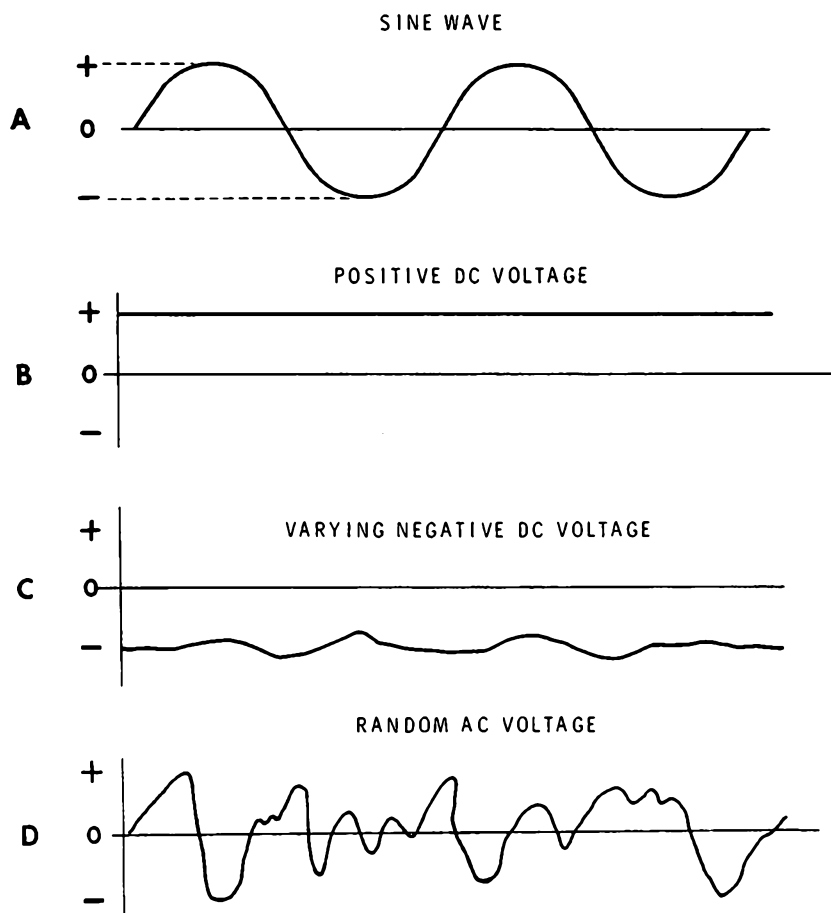


Figure 1-1
Types of Analog signals

Figure 1-1A shows the most common type of analog signal, a sine wave. A significant number of electronic signals are sinusoidal. Radio signals and audio tones are examples. A fixed dc voltage is also an analog signal. Figure 1-1B shows a constant positive dc voltage. Another type of analog signal is a varying dc voltage or current. A changing negative dc voltage is illustrated in Figure 1-1C. Any random but continuously varying voltage waveform is considered to be analog. The signal shown in Figure 1-1D is only one of an infinite variety of such signals. Electronic circuits that process these analog signals are called linear circuits.

Digital signals are essentially a series of pulses or rapidly changing voltage levels that vary in discrete steps or increments. Digital signals are pulses of voltage that usually switch between two fixed levels. Figure 1-2 shows several types of digital signals. Notice how these signals switch between two distinct voltage levels. In Figure 1-2A, the two levels are 0 (ground) and +5 volts. In Figure 1-2B, the levels are 0 (ground) and -6 volts. In Figure 1-2C the signal alternates between the +3 and -3 volt levels. This two-level, off-on or up-down fast switching characteristic is fundamental of all digital signals. Electronic circuits that process these digital signals are called digital, logic, or pulse circuits.

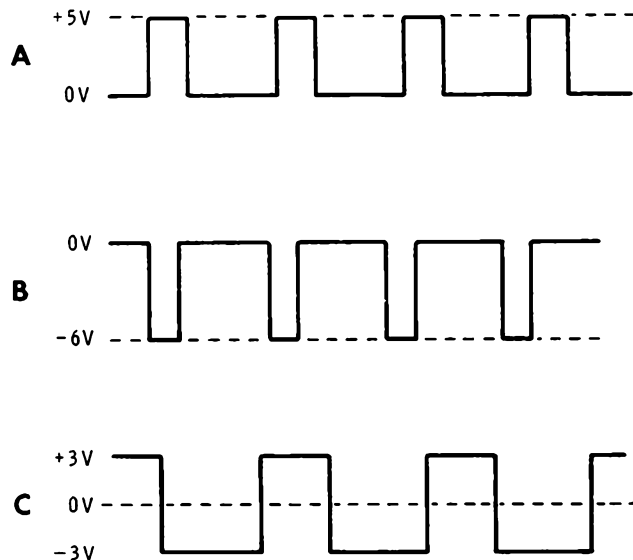


Figure 1-2
Types of Digital signals

Contrasting Analog and Digital Devices and Techniques

Now let's further define analog and digital methods in terms of devices and ideas that are already quite familiar to you. For example, a light bulb can be either an analog or digital device depending upon how it is used. The amount of current through a light bulb can be set to any level less than its maximum rated value. We can vary the current through it continuously and its brightness will vary. Used in this way the light bulb is an analog device. The brilliance of the lamp is proportional to the current through it. There are virtually an infinite number of brightness levels.

The lamp can also be used as a digital device where the current through it and its brightness varies in discrete steps. The most common way of using the light bulb as a digital device is to give it two brilliance levels, usually off and on. The important point is that the lamp has two states. Because of this off-on characteristic, we say that the lamp is binary in nature. The term binary designates any two-state device or signal.

Let's take some other examples to illustrate the concept of analog and digital techniques. The VHF channel selector switch on your television set is digital in nature because it can assume only discrete positions. It can be set to any one of thirteen unique states, channels 2 through 13 and UHF. Any type of switch is a digital device because it has two or more discrete positions.

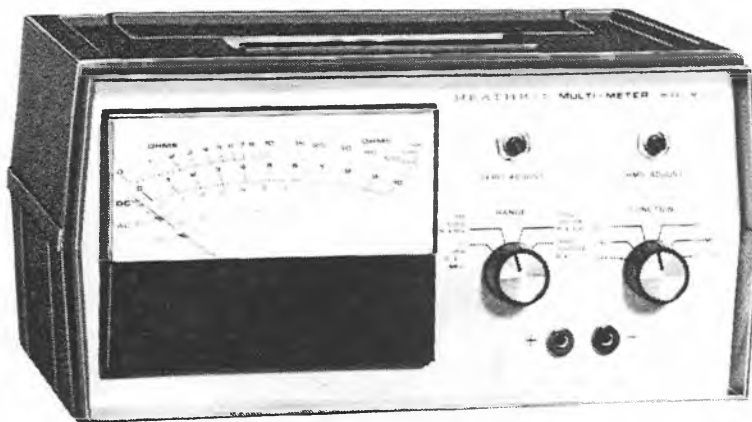
The volume control on your television set is an analog device. You can vary the volume of the sound continuously over a wide range from completely off to extremely loud.

The speedometer on your car is an analog device. It tells you the speed of your car in miles per hour on a smooth and continuously varying basis. You read the speed from a dial that is usually calibrated in no smaller increments than 5 miles per hour. To determine the speeds between the markings you must interpolate, or guess at, the exact speed.

The odometer portion of your speedometer, the part that indicates the number of miles traveled, is a digital device. Since the odometer records mileage in increments of one mile (or in some cases one-tenth mile), it is digital in nature.

Another example of an analog device is a typical clock or watch. It indicates the time continuously by the positions of the hands on a calibrated dial. The second hand sweeps smoothly and continuously around in an analog fashion as do both hour and minute hands. To determine the exact time, you must estimate the positions of the hands. Your ability to read the time accurately is limited by the precision of the dial calibration increments. Digital clocks overcome this problem. On a digital clock, you read the time directly from decimal number display readouts in discrete increments of hours, minutes and seconds. The accuracy is greatly improved and you gain the added convenience of a direct number display.

A standard voltmeter is also an analog device. It reads or measures voltage and indicates its value by the position of a pointer on a meter scale. The pointer moves smoothly or continuously as the amplitude of the analog voltage being measured varies. Of course, digital voltmeters are also available. These instruments measure the voltage and display it as discrete digits on a decimal readout.



An analog multimeter for measuring voltage, current and resistance



A typical digital multimeter

Here are a few other analog quantities and devices:

temperature — thermometer

direction — compass

light intensity — light meter

Keep in mind that all of these variables could be monitored and displayed as a digital readout.

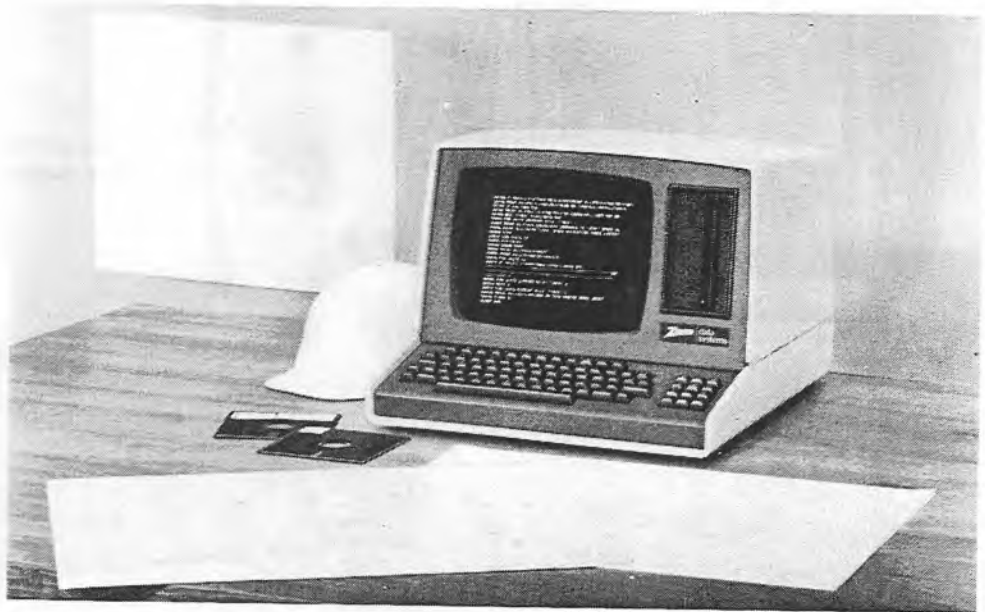
Further examples of digital variables are:

money

heart pulse rate

Where Are Digital Techniques Used?

Perhaps the greatest use of digital techniques today is in computers. Digital computers are used in virtually all areas of business and industry. They are extremely useful machines that can save man a tremendous amount of effort and greatly extend his capabilities



A typical microcomputer.

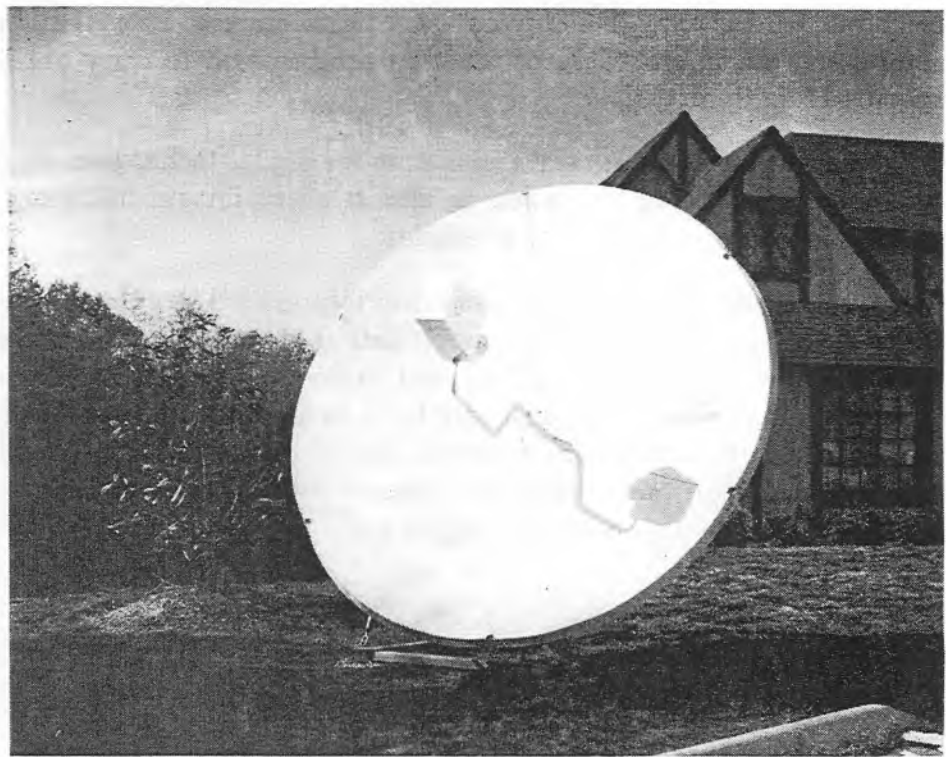
Over the years digital computers have grown in capability, but have become smaller, cheaper and easier to use. As a result, their use has increased tremendously. The small, low-cost but very powerful minicomputer has put digital and computer techniques within the reach of nearly everyone. A microcomputer can be quickly and easily designed into a system to replace more conventional equipment and circuitry for control, computation and automation.

Advanced semiconductor technology has given us a computer that is really a component. Known as microprocessors, these devices are complete digital computers in a single miniature integrated circuit package. Microprocessors can replace minicomputers in many applications and can often be used to replace conventional digital circuits. Like all digital computers, the microprocessor must be programmed to carry out its specified function.

But computers aren't the only application for digital techniques. Digital methods are being employed in almost every imaginable area of electronics. Here are just a few examples.

Communications. Instead of transmitting information over wire lines or by radio by analog methods, much data is now being transmitted in digital form. It has been found that pulse type signals are easier to work with and are less susceptible to noise and other problems common in communications systems. Digital computers can readily communicate with one another by transmitting information over the telephone lines by using digital techniques.

Telemetry systems. Those systems, used for transmitting measurement data from a remote location, use digital techniques extensively. In an unmanned satellite, for example, sensors are used to monitor various environmental conditions such as temperature, light, and radiation. The analog voltages produced by these sensors, in response to the quantity being measured, can be transmitted back to the earth via radio by modulating a carrier using conventional analog methods. However, it has been found that by converting the analog variables into digital signals, an improvement in transmission reliability and accuracy can be achieved. Today, telemetry systems in satellites and missiles make extensive use of digital techniques.



A three-meter satellite antenna.

Test Instruments. The trend in test and measurement equipment is clearly toward the use of digital techniques. Besides the convenience of a direct decimal display and the increased precision of measurement brought about by the use of digital techniques, it is possible to interface many digital instruments with a computer. This permits automatic monitoring, controlling, measuring and recording of data.

The most common electronic test instrument, the analog voltmeter, is gradually being replaced by the more sophisticated digital voltmeter (DVM). The DVM does the same job as its analog counterpart. The DVM measures voltage but instead of presenting the reading to the observer in the form of a pointer on a meter face, the voltage is a direct readout display of decimal numbers. Such an instrument, while generally higher in cost than an analog voltmeter, is extremely convenient to use and read. More important, it gives more accurate measurements of voltages. The digital multimeter (DMM), a DVM with refinements, is capable of measuring voltages as well as resistance and current like an analog VOM.

Another widely used digital test instrument is the electronic counter. This unit is widely used for measuring frequency and time intervals. Again, digital techniques provide a convenient decimal read-out of the exact quantity being measured, thereby eliminating man's need to interpolate continuous or analog meter scales to provide a reading. This result is greater accuracy and less error in measurement.



A digital counter used for measuring frequency.

Industrial Controls. Digital techniques are becoming more widely used in manufacturing plants and refineries where complex operations must be accurately controlled. These systems use sensors to monitor various phases of the operation, and the outputs of these sensors are then used to produce signals that will control the various operations that affect the process.

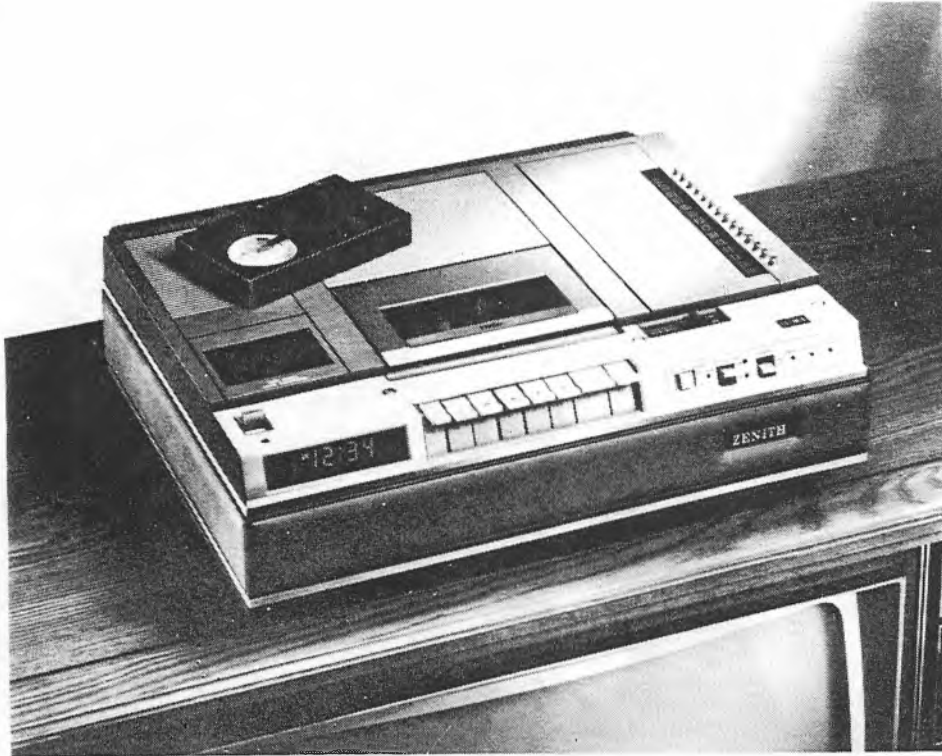
Digital computers are the heart of the new Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) systems that are being more widely used in industry.

Industrial feedback control systems have traditionally used analog techniques. Today, many of these control systems are digital in nature. Most involve both analog and digital methods. Analog quantities like temperature, pressure, liquid level and flow rate are monitored by sensors to produce an analog signal. These are converted into digital values by analog-to-digital converters. Other system inputs are already digital in nature such as limit switches or sensors of an off-on or go no-go nature.

All of this digital information is fed into a digital computer which monitors the input variables and, according to a predetermined program, generates output signals to control the process.

Another popular industrial application for digital techniques is machine tool control. Here, a programmable controller or digital computer controls the drilling, cutting, punching and stamping of materials to produce specific metal parts accurately and automatically.

Consumer Electronic Equipment. Everyday home electronic products now use digital techniques extensively. TV sets feature digital channel section, remote control, programming, telephone dialing, and sync control. Video cassette recorders (VCRs) and video disks also contain digital circuits for tuning, programming, timing, and media selection. Advanced hi-fi equipment now includes digital methods for tuning via phase-locked loop frequency synthesizers. Even digitally controlled record turntables are available. The future promises even more digital audio equipment. The digital recording and playback of music, for example, will further increase fidelity and lower noise. Such digital hi-fi equipment is just now being introduced.



A Video Cassette Recorder using digital techniques.



An FM tuner using digital techniques for frequency selection and display.

Other popular consumer devices using digital methods are the popular digital clocks, telephone dialers, and electronic calculators. Even a home digital thermometer is available.



Television set with remote unit.

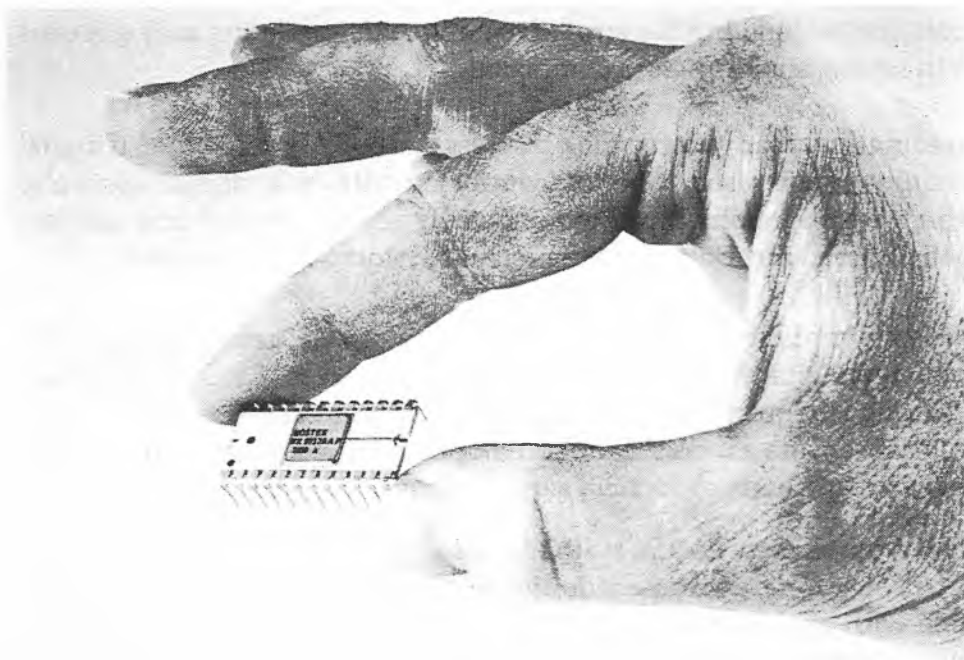


Digital clock.

Why Use Digital Techniques?

The primary reason for the widespread use of digital techniques has been the availability of low cost, digital integrated circuits (ICs). Advances in integrated circuit technology have produced many excellent low cost digital circuits. Such circuits are small, inexpensive and very reliable. Medium scale and large scale (MSI and LSI) integrated devices can replace entire circuits and instruments. Electronic equipment designers recognize the availability of such devices and have begun to take advantage of them. While digital techniques have been known for years, it took integrated circuits to make them practical.

By using digital IC's many equipment improvements have been made. Reductions in size, weight, cost and power consumption usually result when analog techniques are replaced by digital methods. Here are a few more reasons why digital techniques have become so popular.



Greater Accuracy and Resolution. Digital techniques permit greater precision and resolution in representing quantities or in making measurements than with analog methods.

Greater Dynamic Range. Dynamic range is the difference between the upper and lower data values that a system or instrument can handle. Analog systems are limited because of component capabilities and noise to a range of something less than 100,000 to 1. With digital techniques a greatly expanded dynamic range can be obtained.

Greater Stability. Analog or linear circuits are subject to the effects of drift and component tolerance problems. Temperature and other environmental factors affect resistor, capacitor and inductor values. Transistor bias varies causing non-linear operation and distortion to occur. Component imperfections and ageing cause drift and other problems. Digital methods greatly minimize or completely eliminate such problems.

Convenience. Digital techniques make instruments and equipment more convenient to use. The direct decimal display of data is not only more convenient, but the error of reading or interpolating analog meters or in setting analog dials is eliminated.

Automation. Many electronic processes can be fully automated if digital techniques are used. Special control circuits or a digital computer which is programmed can automatically set up, control and monitor many operations. Data is readily recorded, stored and displayed.

Design Simplicity. Digital equipment is relatively easy to design. The available digital ICs make digital design a pleasure. Little or no breadboarding is required. In analog or linear circuits breadboarding is mandatory to ensure a workable circuit. Digital equipment can go from paper design to finished product in a very short time.

New Approaches. Digital methods permit new approaches to the solution of electronic equipment design. In addition, design solutions impossible with analog techniques are readily implemented with digital circuits. Digital circuits make it possible to do some things that have no analog equivalent.

A disadvantage of digital is that it is inherently incompatible with real world variables, which are primarily analog. This necessitates data conversion interfaces, which are complex and costly.

-



Answers

1. continuously (or smoothly), in steps (in discrete increments, etc.)
2. a. analog
b. digital
c. digital
d. digital
e. analog
f. analog
3. Two (2)
4. computers
5. greater accuracy
greater dynamic range
convenience of direct decimal display
and many others.
6. (b) development of integrated circuits
7. (c) a constant dc voltage could be an analog signal or one of two levels in a binary digital system, depending upon the circuits and techniques used or the definitions given.
8. (a) since the waveform has more than two discrete steps, it is considered to be analog.

THE BINARY NUMBER SYSTEM

All digital circuits, instruments, and systems work with numbers that represent specific quantities. For example, the analog voltage measured by a digital voltmeter is converted into digital form and displayed as a specific decimal number. The number that you enter into an electronic calculator is stored and used in the calculation you specify. The digital computer that prints your payroll check works with numbers, specifically your salary, the number of hours you work and the various deductions. As you can see, numbers or quantities are the basic source for an end product of most digital equipment. Figure 1-3 shows how most digital equipment accepts input numbers, processes them and generates number outputs. The actual form of the input and output numbers depends on the application. They may be in a binary or decimal form. In some applications the input and/or output may be in analog form despite the digital processing.

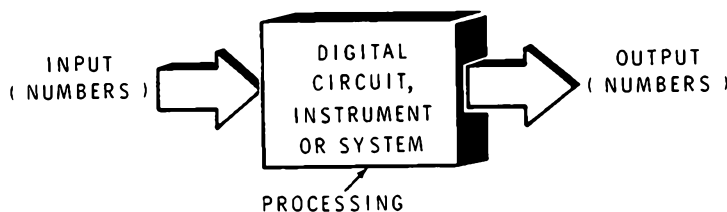


Figure 1-3

A general block diagram of any digital system

The type of numbers we are most familiar with are decimal numbers. In the decimal number system we combine the ten digits 0 through 9 in a certain way so that they indicate a specific quantity. In the binary number system, we use only two digits, 0 and 1. These binary digits, or bits, when appropriately arranged can also represent any decimal number. For example, the binary number 110101 represents the decimal quantity 53. All modern digital techniques are based on the binary number system.

The basic distinguishing feature of a number system is its base or radix. The base indicates the number of characters or digits used to represent quantities in that number system. The decimal number system has a base or radix of 10 because we use the ten digits 0 through 9 to represent quantities. The binary number system has a base of 2 since only the digits or bits 0 and 1 are used in forming numbers.

The decimal number system came about basically as a result of man having ten fingers. Man's earliest attempts to represent numbers, count and keep track of quantities involved the use of his fingers. Of course, the decimal number system is universally used and understood because it is our way of communicating information about quantities.

The binary number system, while simple, is inconvenient to use because we are not familiar with it. But once you learn it, you will find it easy to work with. It has special benefits when it comes to constructing the hardware used in digital equipment.

Digital systems can be implemented with either the decimal or binary number systems. However, the advantages of the binary number systems over the decimal number system in terms of hardware implementation are many. An electronic component or circuit that has only two states is significantly simpler, less expensive, faster and more reliable than one with ten. Each bit can be implemented with components of a simple off/on nature such as switch or relay contacts or a transistor that conducts or does not conduct.

Positional Number Systems

The decimal and binary number systems are positional or weighted number systems. This means that each digit or bit position in a number carries a particular weight in determining the magnitude of that number. For example, you know that a decimal number has positional weights of units, tens, hundreds, thousands, etc. Each position has a weight that is some power of the number system base, in this case ten. The positional weights are $10^0 = 1$ (units)*, 10^1 (tens), 10^2 (hundreds), etc. We evaluate the total quantity represented by considering the specific digits and the weights of their positions. Consider the decimal number 7438 in which there are 8 ones, 3 tens, 4 hundreds, and 7 thousands. The number can be written as indicated below.

$$(7 \times 10^3) + (4 \times 10^2) + (3 \times 10^1) + (8 \times 10^0) = \\ 7000 + 400 + 30 + 8 = 7438$$

To determine the value of the number, you multiply each digit by the weight of its position and add your results.

* Any number with an exponent of zero is equal to one.

Binary numbers work the same way. Each bit position carries a specific weight. As in the decimal number system, the position weights are some power of the base of the number system. These weights from right to left are $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, etc. The weight of each position is twice that of the weight of the number to the right. Consider the binary number 110101. This can be written as indicated below.

$$(1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = \\ 32 + 16 + 0 + 4 + 0 + 1 = 53$$

The quantity represented by the number is determined by multiplying each bit by its position weight and obtaining the sum.

Fractional Numbers

So far we have only discussed whole numbers or integer quantities. But as you know it is often necessary to express quantities in terms of fractional parts of a whole.

Decimal fractions are numbers whose positions have weights that are negative powers of ten such as $10^{-1} = 1/10 = .1$, $10^{-2} = 1/100 = .01$, $10^{-3} = 1/1000 = .001$, etc. A decimal point separates the whole and fractional parts of a number. The integer or whole number portion is to the left of the decimal point and has weights of units, tens, hundreds, etc. The fractional part of the number is to the right of the decimal point and the positions have weights of tenths, hundredths, thousandths, etc. To illustrate this, the number 278.94 can be written as shown below.

$$(2 \times 10^2) + (7 \times 10^1) + (8 \times 10^0) + (9 \times 10^{-1}) + (4 \times 10^{-2}) = \\ 200 + 70 + 8 + .9 + .04 = 278.94$$

In a fractional binary number, the weights of the fractional positions are negative powers of 2 or $2^{-1} = 1/2 = .5$, $2^{-2} = 1/4 = .25$, $2^{-3} = 1/8 = .125$, $2^{-4} = 1/16 = .0625$, etc. The position weight is one half of the weight of the position directly to the left. A binary point separates the whole and fractional parts of the number.

The binary number 1101.101 is evaluated as shown below.


$$(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) = \\ 8 + 4 + 0 + 1 + .5 + 0 + .125 = 13.625$$

Converting Between The Binary and Decimal Number Systems

In working with digital equipment, you will often need to determine the decimal value of binary numbers. In addition, you will also find it necessary to convert a specific decimal number into its binary equivalent. Let's see how such conversions are accomplished.

Binary to Decimal. To convert a binary number into its decimal equivalent you simply add together the weights of the positions in the number where binary 1's occur. The weights of the integer and fractional positions are indicated below.

INTEGER								FRACTIONAL		
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
128	64	32	16	8	4	2	1	.5	.25	.125

Binary Point 

As an example, let's convert the binary number 1010 into its decimal equivalent. Since no binary point is shown, the number is assumed to be a whole number where the binary point is to the right of the number. The right-most bit, called the least significant bit or LSB, has the lowest integer weight of $2^0 = 1$. The left-most bit is the most significant bit (MSB) because it carries the greatest weight in determining the value of the number. In this example, it has a weight of $2^3 = 8$. To evaluate the number we add together the weights of the positions where binary 1's appear. In this example, 1's occur in the 2^3 and 2^1 positions. The decimal equivalent is ten.

Binary Number	1	0	1	0				
Position Weights	(8)	(4)	(2)	(1)				
Decimal Equivalent	8	+	0	+	2	+	0	= 10

As a further illustration of this process, let's convert the binary number 101101.11 into its decimal equivalent.

Binary Number	1	0	1	1	0	1	.1	1								
Position Weights	(32)	(16)	(8)	(4)	(2)	(1)	(.5)	(.25)								
Decimal Equivalent	32	+	0	+	8	+	4	+	0	+	1	+	.5	+	.25	= 45.75

Note that you can disregard the position weights where binary 0s occur since they add nothing to the number value.

After you solve a few practice problems, you will quickly catch on to this procedure.

Decimal to Binary. Converting a known decimal number into its binary equivalent can be accomplished by a simple trial and error method once you are familiar with the weighting sequence of binary numbers. Suppose that you wish to convert the decimal number 175 into its binary equivalent. To do this you first determine the highest positional weight that is equal to or less than the number being converted. This is 128. Subtract 128 from 175 and note the remainder.

$$\begin{array}{r} 175 \\ - 128 \\ \hline 47 \end{array}$$

Again determine the highest positional weight that does not exceed the remainder. This is 32. Next determine their difference and continue to repeat the process until no further subtractions are possible.

$$\begin{array}{r} 47 \\ - 32 \\ \hline 15 \end{array}$$

The highest positional weight less than 15 is 8.

$$\begin{array}{r} 15 \\ - 8 \\ \hline 7 \end{array}$$

The highest positional weight less than 7 is 4.

$$\begin{array}{r} 7 \\ - 4 \\ \hline 3 \end{array}$$

The highest positional weight less than 3 is 2.

$$\begin{array}{r} 3 \\ - 2 \\ \hline 1 \end{array}$$

And finally, the highest positional weight less than or equal to 1, of course, is 1.

Now using this information you can construct the equivalent binary number. You write a binary 1 for the weight positions you subtracted from the original number and the remainders. In this example, these were 128, 32, 8, 4, 2 and 1. Note that you did not use the 64 and 16 weights, so these positions will be binary 0. The number then is $10101111 = 175$. You can check it by converting the binary version back into decimal form using the procedure discussed earlier.

Another method for converting decimal numbers into binary is to repeatedly divide the number by 2 and note the remainder. When dividing by 2, the remainder will always be either 1 or 0. The remainder forms the equivalent binary number.

As an example, convert the number 175 into its binary equivalent.

	REMAINDER
$175 \div 2 = 87$	1 ← LSB
$87 \div 2 = 43$	1
$43 \div 2 = 21$	1
$21 \div 2 = 10$	1
$10 \div 2 = 5$	0
$5 \div 2 = 2$	1
$2 \div 2 = 1$	0
$1 \div 2 = 0$	1 ← MSB

Binary Number Sizes

Binary numbers are also referred to as binary words. An 8 bit binary number is also an 8 bit word. You will also see the term byte used to refer to binary words. Most digital circuits and equipment use a fixed word size. The size of this word determines the maximum magnitude and resolution with which numbers can be represented. The number of bits in a word determine the number of discrete states that can exist and the maximum decimal number value that can be represented.

The formula below indicates the number of states that can be represented with a given number of bits.

$$N = 2^n \quad \begin{array}{l} N = \text{total number of states.} \\ n = \text{number of bits in the word.} \end{array}$$

For example, with a 4-bit word, we can represent a maximum of

$$N = 2^n = 2^4 = 2 \times 2 \times 2 \times 2 = 16$$

This means that by using 4-bit positions, we can create a total of 16 different binary bit patterns or number combinations. These are shown in Table I along with their decimal equivalents.

TABLE I

DECIMAL	BINARY
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Binary and decimal number equivalents for a 4-bit word.

As Table I indicates, we represent the numbers 0 through 15 using the binary number weighting system. The maximum decimal number that can be represented is one less than the total number of states. The largest decimal number value (N) that can be represented for a given number of bits (n) is expressed with the formula below.

$$N = 2^n - 1$$

For example, with a 6-bit number we can represent a maximum value of

$$N = 2^6 - 1 = 64 - 1 = 63$$

If you know the maximum decimal quantity (N) that you wish to represent with a binary number, you can determine the required number of bits (B) with the expression given below.

$$B = 3.32 \log_{10}(N + 1)$$

The common logarithm can be obtained from a set of tables, a slide rule or an electronic calculator with log capability. For example, if the maximum decimal number that you need to represent is 500, the number of bits required is:

$$B = 3.32 \log_{10} 500$$

$$B = 3.32 (2.6998377)$$

$$B = 8.96$$

Of course, you cannot implement fractional bits so the total number of bits required is the next highest whole number. Therefore, to represent the number 500, you would need a total of nine bits. Using the previously given expression, you can determine that, with a total of nine bits, the maximum number you can represent is

$$N = 2^n - 1 = 2^9 - 1 = 512 - 1 = 511$$

The Appendix at the end of this unit contains a table of numbers that are powers of 2. It will help you to quickly determine the relationship between decimal number size and binary word bit length.

Number Identification

When working with both binary and decimal numbers, it is often necessary to have some way of identifying whether a number is a decimal or a binary number. This is particularly true of numbers involving only ones and zeros. For example, the number 101 could represent a quantity of one hundred and one if it is a decimal number. However, if this number is in binary form, it would represent a quantity of five.

To distinguish binary from decimal numbers a small subscript number is generally written after the number. The identifying subscript number is the base or radix of the number system being used. Several examples of this notation are indicated below.

$$101_2 = 5_{10}$$

$$101_{10} = 1100101_2$$

Self Test Review

9. The radix of the binary number system is _____ .
10. Binary hardware is preferred over decimal hardware in digital equipment because binary components are
 - a. _____
 - b. _____
 - c. _____
11. Convert the following binary numbers to decimal.
 - a. 100101101
 - b. 11100.1001
 - c. 111111
 - d. 100000.0111
12. Convert the following decimal numbers to binary.
 - a. 127
 - b. 38
 - c. 22.
 - d. 764.
13. What is the largest decimal number that can be represented with 8 bits?
14. How many discrete states can be represented with 6 bits?
15. How many bits does it take to represent the number 3875 in binary?

Answers

- 9. 2
- 10. a. faster
 - b. simpler
 - c. more reliable
- 11. a. $100101101_2 = 301_{10}$
 - b. $11100.1001_2 = 28.5625_{10}$
 - c. $111111_2 = 63_{10}$
 - d. $100000.0111_2 = 32.4375_{10}$
- 12. a. $127_{10} = 1111111_2$
 - b. $38_{10} = 100110_2$
 - c. $22_{10} = 10110_2$
 - d. $764_{10} = 1011111100_2$
- 13. $N = 2^8 - 1 = 256 - 1 = 255$
- 14. $N = 2^6 = 64$
- 15. $B = 3.32 \log_{10} 3875 = 3.32 (3.58827) = 11.91$ or 12 bits

BINARY CODES

The general term given for the process of converting a decimal number into its binary equivalent is coding. We can express a decimal number as an equivalent binary code or binary number. The binary number system, as discussed, is known as the pure binary code. We give it this name to distinguish it from other types of binary codes. In this section you will see some of the other types of binary codes used in digital systems.

Binary Coded Decimal

Because the decimal number system is so familiar, it is easy to use. The binary number system is less convenient to use because we are not as intimately familiar with it. It is difficult to quickly glance at a binary number and recognize its decimal equivalent. For example, the binary number 1010011 represents the decimal number 83. You certainly cannot tell immediately by looking at the number what its decimal value is. However, you know that within a few minutes, using the procedures described earlier, that you could readily calculate its decimal value. The amount of time that it takes you to convert or recognize a binary number quantity is a distinct disadvantage in working with this code despite the numerous hardware advantages. Digital engineers recognized this problem early and developed a special form of binary code that was more compatible with the decimal system. Because so many digital devices, instruments and equipment use decimal input and output, this special code has become very widely used and accepted. This special compromise code is known as binary coded decimal (BCD). The BCD code combines some of the characteristics of both the binary and decimal number systems.

The BCD code is a system that represents the decimal digits 0 through 9 with a four-bit binary code. This BCD code uses the standard 8421 position weighting system of the pure binary code. The standard 8421 BCD code and the decimal equivalents are shown in Table II. As with the pure binary code, you can convert the BCD numbers into their decimal equivalents by simply adding together the weights of the bit positions where the binary 1's occur. Since 9 is the largest character used as a digit in the decimal number system, the 4-bit binary numbers representing the decimal numbers 10 through 15 are invalid in the BCD system.

TABLE II
8421 BCD CODE

DECIMAL	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

To represent a decimal number in BCD notation you simply substitute the appropriate four-bit code for each decimal digit. For example, the number 834 in BCD would be 1000 0011 0100. Each decimal digit is represented by its equivalent 8421 four-bit code. A space is left between each four-bit group in order to avoid confusing the BCD format with the pure binary code.

The beauty of the BCD code is that the ten BCD code combinations are very easily remembered. Once you begin to work with binary numbers on a regular basis, you will find that the BCD numbers will come to you as quickly and automatically as decimal numbers. For that reason, by simply glancing at the BCD representation of a decimal number you can make the conversion almost as quickly as if it were already in decimal form.

While the BCD code does help to simplify the man-machine interface it is less efficient than the pure binary code. It takes more bits to represent a given decimal number in BCD than it does with pure binary notation. For example, the decimal number 83 in pure binary form is 1010011. In BCD code the decimal number 83 is written as 1000 0011. In the pure binary code it takes only a 7-bit word to represent the number 83. In BCD form it takes 8 bits. The inefficiency arises from the fact that for each bit in a data word there is usually a certain amount of digital circuitry associated with it. The extra circuitry associated with the BCD code costs more, increases equipment complexity, and consumes more power. Arithmetic operations with BCD numbers are also more time-consuming and complex than those with pure binary numbers. As you recall, with four bits of binary information we can represent a total of $2^4 = 16$ different states or the decimal number equivalents 0 through 15. In the BCD system we waste six of these states (10-15), thus compounding the inefficiency. Therefore, when we use the BCD number system, we trade off some efficiency for the improved communications between the digital equipment and the human operator.

Special Binary Codes

Besides the standard pure binary coded form, the BCD numbering system is by far the most widely used digital code. You will find one or the other in most of the applications that you encounter. However, there are several other codes that are used for special applications. Let's consider some of these.

Gray Code

The Gray Code is a widely used non-weighted code system. Also known as the cyclic, unit distance or reflective code, the Gray code can exist in either the pure binary or BCD formats. The Gray code is shown in Table III. As with the pure binary code, the first ten codes are used in BCD operations. Notice that there is a change in only one bit from one code number to the next in sequence. You can get a better idea about the Gray code sequence by comparing it to the standard four-bit 8421 pure binary code also shown in Table III. For example, consider the change from 7 (0111) to 8(1000) in the pure binary code. When this change takes place all bits change. Bits that were 1's are changed to 0's and 0's are changed to 1's. Now, notice the code change from 7 to 8 in the Gray code. Here 7(0100) changes to 8 (1100). Only the first bit changes.

TABLE III
THE GRAY CODE

DECIMAL	GRAY	PURE BINARY
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

The Gray code is generally known as an error-minimizing code because it greatly reduces the possibility of ambiguity in the electronic circuitry when changing from one state to the next. When binary codes are implemented with electronic circuitry, it takes a finite period of time for bits to change from 0 to 1 or 1 to 0. These state changes can create timing and speed problems. This is particularly true in the standard 8421 codes where many bits change from one combination to the next. When the Gray code is used, however, the timing and speed errors are greatly minimized because only one bit changes at a time. This permits code circuitry to operate at higher speeds with fewer errors.

The biggest disadvantage of the Gray code is that it is difficult to use in arithmetic computations. Where numbers must be added, subtracted or used in other computations, the Gray code is not applicable. In order to perform arithmetic operations the Gray code number must generally be converted into pure binary form.

ASCII Code

The American Standard Code for Information Exchange (ASCII) code is a special form of BCD code that is widely used in digital computers and data communications equipment. It is a 7-bit binary code that is used in transferring data between computers and their external peripheral devices and in communicating data by radio and telephone lines. With 7 bits we can represent a total of $2^7 = 128$ different states or characters. The ASCII code is used to represent the decimal numbers 0 through 9, the letters of the alphabet (both upper and lower case), and other special characters such as punctuation marks and codes that are used to control various computer peripheral devices and communication circuits. The standard ASCII code is shown in Table IV.

TABLE IV
AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

COLUMN		0	1	2	3	4	5	6	7
ROW	BITS 4321 765	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	\	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
10	1010	LF	SUB	*	:	J	Z	j	z
11	1011	VT	ESC	+	;	K	[k	{
12	1100	FF	FS	,	<	L	\	l	
13	1101	CR	GS	-	=	M]	m	}
14	1110	SO	RS	.	>	N	^	n	~
15	1111	SI	US	/	?	O	_	o	DEL

Explanation of special control functions in columns 0, 1, 2 and 7.

NUL	Null	DLE	Data Link Escape
SOH	Start of Heading	DC1	Device Control 1
STX	Start of Text	DC2	Device Control 2
ETX	End of Text	DC3	Device Control 3
EOT	End of Transmission	DC4	Device Control 4
ENQ	Enquiry	NAK	Negative Acknowledge
ACK	Acknowledge	SYN	Synchronous Idle
BEL	Bell (audible signal)	ETB	End of Transmission Block
BS	Backspace	CAN	Cancel
HT	Horizontal Tabulation (punched card skip)	EM	End of Medium
LF	Line Feed	SUB	Substitute
VT	Vertical Tabulation	ESC	Escape
FF	Form Feed	FS	File Separator
CR	Carriage Return	GS	Group Separator
SO	Shift Out	RS	Record Separator
SI	Shift In	US	Unit Separator
SP	Space (blank)	DEL	Delete

The 7-bit ASCII code for each number, letter or control function is made up of a 3-bit group followed by a 4-bit group. Figure 1-4 shows the arrangement of these two groups and the numbering sequence. The 3-bit group is on the left and bit 1 is the LSB. Note how these groups are arranged in rows and columns in Table IV.

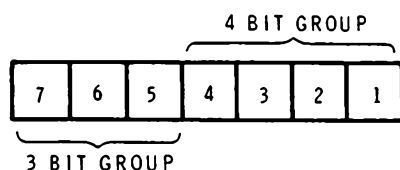


Figure 1-4
ASCII Code Word Format

To determine the ASCII code for a given number, letter, or control operation, you locate that item in the table. Then you use the three and four-bit codes associated with the column and row in which the item is located. For example, the ASCII code for the letter L is 1001100. It is located in column 4, row 12. The most significant 3-bit group is 100, while the least significant four-bit group is 1100. The complete code is 1001100.

There are both 6 and 8-bit special versions of the ASCII code. In addition, the International Business Machines Corporation (IBM) uses another 8-bit coding system called Extended Binary Coded Decimal Interchange Code (EBCDIC) instead of ASCII, for its peripheral and data communications operations.

Self Test Review

16. The BCD code is better than the binary code because
 - a. it uses less bits.
 - b. it is more compatible with the decimal number system.
 - c. it is more adaptable to arithmetic computations.
 - d. there are more different coding schemes available.
17. Convert the following decimal numbers to 8421 BCD code.
 - a. 1049
 - b. 267
 - c. 835
18. Convert the following 8421 BCD code numbers to decimal.
 - a. 1001 0110 0010
 - b. 0111 0001 0100 0011
 - c. 1010 1001 1000
 - d. 1000 0000 0101
19. Which code is best for minimum hardware errors?
 - a. ASCII
 - b. 8421 BCD
 - c. pure binary
 - d. Gray
20. Which BCD code is used for data communications?
 - a. Gray
 - b. 8421
 - c. ASCII
21. The ASCII code is used primarily in _____ and _____ .
22. What is the ASCII code for the letter "f"? _____ .

Answers

- 16. b. more compatible with the decimal system.
- 17. a. 0001 0000 0100 1001
 - b. 0010 0110 0111
 - c. 1000 0011 0101
- 18. a. 962
 - b. 7143
 - c. invalid (1010)
 - d. 805
- 19. d. Gray
- 20. c. ASCII
- 21. computers, data communications.
- 22. 1100110

DATA REPRESENTATION

Now that you understand the reason for using the binary number system and are familiar with some of the binary coding schemes used in digital equipment, you are ready to consider the actual hardware means of implementing these binary numbers. By hardware we mean the electronic components and circuits that are used to represent and manipulate the binary numbers used in the digital system. It is relatively easy to represent a binary number with electronic components. The component, to represent a specific bit in a binary word, must be capable of assuming two distinct states. One of the states will represent a binary 0 and the other a binary 1.

Electromechanical Devices

Switches and relays are ideal for representing binary data. A closed switch or relay contact can represent a binary 1 while the open switch or contact can represent a binary 0. Of course these logic representations can also be reversed. Switches and relays are still widely used to implement digital systems or parts of digital equipment. They are used in places where static binary conditions are required or very low speed operation can be tolerated.

Early digital equipment such as computers and test instruments used relays to represent binary numbers. But the relays were soon replaced by vacuum tubes in many applications. Each bit was represented by a vacuum tube that was either conducting or cut-off. When the tube was conducting it represented one binary state and when it was cut-off it represented the other binary state. Vacuum tubes worked well in digital applications. They achieved operating speeds significantly higher than that of relays. However, because of their large size, high power consumption and speed limitations, they have been replaced by solid state devices.

Transistors

Today the most common way of representing binary data in digital equipment is with a transistor. A transistor can readily assume two distinct states, conducting and cut-off. When a transistor is cut-off it is essentially an open circuit. When a transistor is conducting heavily, it acts as a very low resistance and accurately simulates a closed switch. Most digital circuitry in use today uses saturated bipolar switching transistors for data representation. Non-saturated bipolar transistor switches are also used in many applications where high speed operation is desirable. Keep in mind that both discrete component and integrated circuit transistors are used in digital applications.

The enhancement mode MOS field effect transistor (MOS FET) is also widely used as a two state switch to represent binary data. This type of transistor is the key element in MOS and CMOS integrated circuits.

Logic Levels

The basic element for representing a single bit of data is a switch: mechanical, electromechanical or electronic. The on-off nature of a switch makes it perfect for binary data representation. The exact relationship between the state of the switch and the bit represented by this switch is arbitrary. In actual digital hardware we are not so much concerned with whether the transistor is off or on. Instead the bit assignments are generally represented by voltage levels. The switching element controls these voltage levels. For example, a binary 0 may be represented by 0 volts or ground. A binary 1 may be represented by +5 volts. Depending upon the equipment power supplies available, the exact circuitry used, and the application, almost any voltage level assignments can be used.

Figure 1-5 shows two ways in which a bipolar transistor can be used to produce two distinct voltage levels. In Figure 1-5A, the transistor is connected as a shunt switch. This means that the transistor is in parallel with the output. When the transistor is not conducting, the output voltage is +5 volts as seen through collector resistor R_c . When the transistor is conducting it acts as a very low resistance or near short circuit. At this time the output is some low positive voltage level near ground or zero volts. The switching of the transistor of course is controlled by the application of the appropriate base signal. Switching times in the nanosecond (10^{-9} seconds) region are possible with modern transistors.

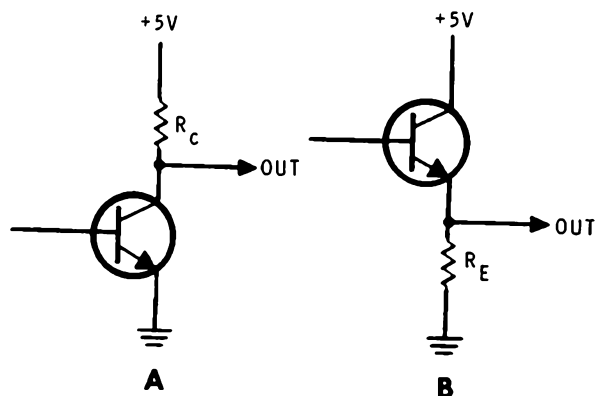


Figure 1-5
Bipolar transistor logic switches, (A)
shunt switch (B) series switch.

In Figure 1-5B the transistor is connected as a series switch. When the transistor is cut off the output is zero volts or ground as seen across resistor R_E . When the transistor conducts, it acts as a very low impedance and connects the +5 volts supply line to the output. Again the operation of the transistor is controlled by applying the appropriate signal to the base. You will find both series and shunt transistor switches used in digital circuits.

Positive and Negative Logic

There are two basic types of logic level representation, positive logic and negative logic. When the most positive of two voltage levels is assigned the binary 1 state, we say that positive logic is being used. When the negative or least positive of two voltage levels is assigned to the binary 1 state, we say that negative logic is being used. Indicated below are several examples of both positive and negative logic level assignments. Keep in mind that the assignments are strictly arbitrary and are selected by the designer when the circuit or equipment is designed.

Positive Logic	Negative Logic
binary 0 = +.2V.	binary 0 = +3.4V.
binary 1 = +3.4V.	binary 1 = +.2V.
binary 0 = -6V.	binary 0 = 0 V.
binary 1 = 0V.	binary 1 = -6V.
binary 0 = +1V.	binary 0 = +15V.
binary 1 = +15V.	binary 1 = +1V.

Parallel vs. Serial Data Representation

There are two basic ways in which digital numbers are transmitted, processed or otherwise manipulated. These methods are designated as serial and parallel. In the serial methods of data handling, each bit of binary word or number is processed serially one at a time. In a parallel system all bits of a word or a number are processed simultaneously.

Serial Data. Figure 1-6 shows a binary number represented in a serial data format. The binary number exists as a series of voltage levels representing the binary 1s and 0s. These voltage level changes occur at a single point in a circuit or on a single line. Each bit of the word exists for a specific interval of time. The time interval allotted to each bit is in this example, one millisecond. The most significant bit (MSB) is the one at the far left. It occurs first since time is considered to be increasing from left to right. Because this is an 8 bit binary word, it takes 8 milliseconds for the entire word to occur or be transmitted. Positive logic level assignments are used. By observing the voltage levels at the specific point or on the transmission line, the number can be determined. The number is 10110010. This is the binary equivalent of the decimal number 178.

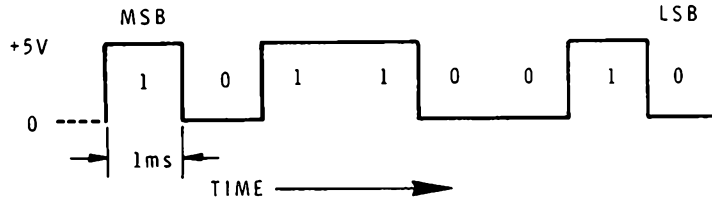


Figure 1-6

The serial binary word 10110010

The primary advantage of the serial binary data representation is that it requires only a single line or channel for transmitting it from one place to another. In addition, since each bit on the single line occurs separately from the others, only one set of digital circuitry is generally needed to process this data. For these reasons, serial data representation is the simplest and most economical of the two types. Its primary disadvantage is that the transmission and processing time required for a serial word is significant since the bits occur one after the other. Despite this time penalty, serial data representation is widely used because of its economy and simplicity.

Parallel Data. The other method of representing, transmitting and processing binary data is designated as parallel. The reason for this is that all bits of a binary word or number are transmitted or processed simultaneously. For this reason a separate line or channel is required for each bit of the word in transmitting that word from one point to another. Refer to Figure 1-7. Here the 8-bit digital word 10110010 is available as voltage levels on eight separate output lines. Since all of the bits of the word are available at the same time, digital circuitry must be provided to process or otherwise manipulate each of the bits in the word simultaneously. The transmission and processing of parallel data, therefore, is more complex and expensive than that required for serial data. However, the clear advantage of parallel data transmission is its speed. All bits are processed at the same time and, therefore, the time required for handling of the data is very short. For high speed applications requiring rapid processing, parallel digital techniques are preferred.

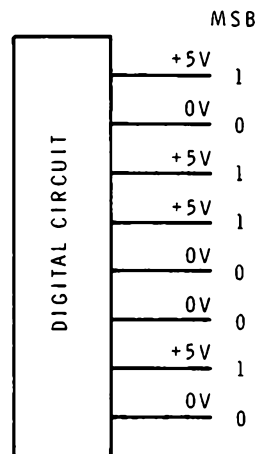


Figure 1-7

The parallel binary
word 10110010.

Self Test Review

23. The basic component used to represent a binary digit is a _____.
24. The two types of transistors used to implement digital circuits are _____ and _____.
25. Designate the following logic level assignments as being either positive or negative.
 - a. binary 0 = +3
binary 1 = -3
 - b. binary 0 = +0.8
binary 1 = +1.8
26. Serial data transmission is faster than parallel data transfers.
 - a. True
 - b. False
27. The following voltage levels appear on six parallel data lines designated A through F.
A = +5 V, B = +5 V, C = 0 V, D = +5 V, E = 0 V, F = +5 V
Using positive logic and assuming bit A is the LSB, what is the decimal number equivalent?

Answers

23. switch (mechanical, electro-mechanical or electronic)
24. bipolar, MOS FET
25. a. negative
 b. positive
26. b. False
27. $A = +5 \text{ V} = 1$ (LSB), $B = +5 \text{ V} = 1$, $C = 0 \text{ V} = 0$, $D = +5 \text{ V} = 1$, $E = 0 \text{ V} = 0$, $F = +5 \text{ V} = 1$ (MSB)
 Number = FEDCBA = 101011_2
 Decimal equivalent = 43_{10}

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and answer all of the questions first. Then check your work against the correct answers in the "Examination Answers."

1. Designate whether each item below is analog or digital:
 - a. auto gasoline gauge _____
 - b. stepladder _____
 - c. blood pressure indicator _____
 - d. radio dial _____
 - e. porch light _____
2. Which of the following is **not** an advantage of digital over analog methods?
 - a. automation
 - b. lower cost interfaces
 - c. design simplicity
 - d. increased dynamic range
 - e. increased resolution
3. Name five consumer electronic products that use digital techniques.
 - a. _____
 - b. _____
 - c. _____
 - d. _____
 - e. _____
4. The signal shown in Figure 1-8 is:
 - a. digital
 - b. analog

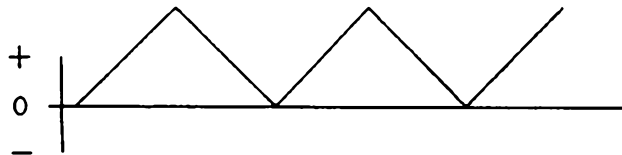


Figure 1-8

Signal for Question 4

5. The increased use of digital techniques resulted primarily from:
 - a. military electronic needs
 - b. the desire to reduce costs
 - c. the development of integrated circuits
 - d. limitations of analog methods
6. Convert the following binary numbers to decimal.
 - a. 11010010
 - b. 10110
7. Convert the following decimal numbers to binary.
 - a. 101
 - b. 62
8. Convert the following BCD numbers to decimal.
 - a. 0001 1001 1000 0101
 - b. 0101 0100 0011 1000
9. Convert the decimal numbers below to BCD.
 - a. 260
 - b. 531
10. In BCD, the code 1011 is:
 - a. 11
 - b. letter B
 - c. 23
 - d. invalid
11. a. The highest number that can be represented with 12 bits is _____
b. It takes a minimum of _____ bits to represent the number 675.
12. The digital code for best man-machine communications is _____
13. The digital code used in transmitting data between computers and their peripherals is _____
14. The digital code used to minimize errors in data conversion is _____
15. Designate which of the following is positive or negative logic.
 - a. binary 1 = -.8
binary 0 = -1.8 _____
 - b. binary 1 = 0 volts
binary 0 = +10 volts _____

16. Refer to Figure 1-9 and answer the following questions:

- a. The output from this logic circuit is:
 - (1) serial
 - (2) parallel
- b. If binary 0 = 0 volts and binary 1 = +5 volts, the decimal equivalent of the binary output is _____.

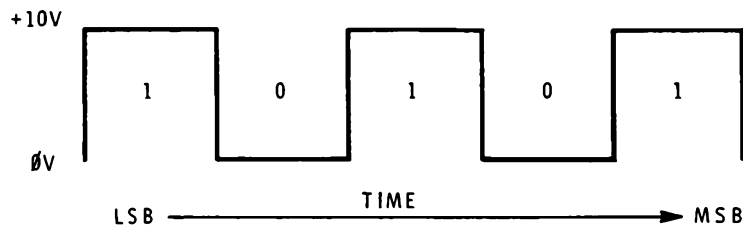


Figure 1-9

Illustration for Question 16

17. The two types of electronic components most often used to generate binary signals are _____ and _____.
18. The word used to designate an 8-bit binary number is:
 - a. nibble
 - b. byte
 - c. word
 - d. code
19. Draw the serial data waveform for the ASCII Code of the letter J. (Use Table IV.) The MSB should be first in time, and Binary 1 = +5 volts, Binary 0 = 0 volts.

20. From the list below, select the main advantage and disadvantage of serial and parallel data transmission and write the letter into the designated spaces.

- a. high speed
- b. slow speed
- c. maximum cost/complexity
- d. minimum cost/complexity

<u>Serial</u>	<u>Parallel</u>
Advantage _____	Advantage _____
Disadvantage _____	Disadvantage _____

EXAMINATION ANSWERS

1. a.—auto gasoline gauge - analog
b.—stepladder - digital
c.—blood pressure indicator - analog
d.—radio dial - analog
e.—porch light - digital
2. b.—lower cost interfaces. Complex, costly interfaces are normally used between digital equipment and the normal analog world.
3. a.—TV set
b.—VCR
c.—stereo receiver
d.—digital clock
e.—electronic calculator
4. b.—analog. The smoothly varying continuous nature of the signal makes it analog.
5. c.—Development of the integrated circuit is the main reason for the rapid increase in the use of digital techniques.
6. a.— $11010010_2 = 210_{10}$
b.— $10110_2 = 22_{10}$
7. a.— $101_{10} = 1100101_2$
b.— $62_{10} = 111110_2$
8. a.— $0001\ 1001\ 1000\ 0101 = 1985$
b.— $0101\ 0100\ 0011\ 1000 = 5438$
9. a.— $260 = 0010\ 0110\ 0000$
b.— $531_{10} = 0101\ 0011\ 0001_2$
10. d.—invalid. 1011 is not a BCD value.
11. a.— $M = 2^{12} - 1 = 4096 - 1 = 4095$
b.— $B = 3.32 \log_{10} 675 = 3.32 (2.8293) = 9.3932$ or 10 bits
12. BCD is best for man-machine communications.
13. ASCII
14. Gray
15. a.—Binary 1 = -.8 volts
Binary 0 = -1.8 volts positive logic
b.—Binary 1 = 0 volts
Binary 0 = +10 volts negative logic
16. a.—(1) Serial
b.—The binary output number is 10101 or 21_{10} .

17. Switches, transistors
18. b.-byte, an 8 bit number.
19. See Figure 1-10

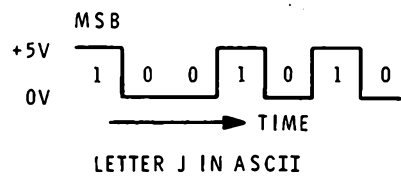


Figure 1-10

Answer for Question 19.

20. Serial

Advantage - D. minimum cost/complexity

Disadvantage - B. slow speed

Parallel

Advantage - A. high speed

Disadvantage - C. maximum cost/complexity

APPENDIX

Table of Powers of 2

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5

n = number of bits

Unit 2

**SEMICONDUCTOR
DEVICES FOR
DIGITAL CIRCUITS**

CONTENTS

Introduction	2-3
Unit Objectives	2-4
Unit Activity Guide	2-5
The Bipolar Transistor Switch	2-6
Designing a Saturated Switch Logic Inverter	2-19
MOS Field Effect Transistors	2-28
Experiment 1 — Bipolar Transistor Switch	2-35
Unit Examination	2-41
Examination Answers	2-44

INTRODUCTION

At the heart of all modern digital circuits are semiconductor devices like diodes and transistors. Your ability to understand digital circuits and apply them to practical situations depends directly on a knowledge of semiconductors. The purpose of this Unit is to provide you with a solid background in semiconductor fundamentals as they apply to digital circuits, both discrete components as well as integrated circuits. Our discussion here will focus on practical operation and application rather than detailed coverage of internal device physics.

The “Unit Objectives” will tell you exactly what you will learn in this Unit. Follow the “Unit Activity Guide,” checking off each learning activity as you complete it. Keep track of your time and progress in the spaces provided.

UNIT OBJECTIVES

When you complete Unit 2 on semiconductor devices, you will have the knowledge and skills indicated below. You will be able to:

1. Name the two types of semiconductor elements used in digital circuits and list the advantages and disadvantages of each.
2. Identify from a list the symbols used to represent PNP and NPN bipolar transistors and P- and N-channel enhancement mode MOSFETs.
3. Explain the operation of both bipolar transistors and MOSFETs.
4. Name and explain the three operating modes of a bipolar transistor.
5. Determine the proper bias on a bipolar transistor for saturated operation.
6. Explain the operation of a logic inverter circuit.
7. Design a saturated bipolar transistor switching circuit.

UNIT ACTIVITY GUIDE

	Completion Time
<input type="checkbox"/> Read "The Bipolar Transistor Switch."	_____
<input type="checkbox"/> Answer Self Test Review Questions 1-6.	_____
<input type="checkbox"/> Read "Switching Speed."	_____
<input type="checkbox"/> Answer Self Test Review Questions 7-10.	_____
<input type="checkbox"/> Read "Designing a Saturated Switch Logic Inverter."	_____
<input type="checkbox"/> Work Self Test Review problem 11.	_____
<input type="checkbox"/> Read "MOS Field Effect Transistors."	_____
<input type="checkbox"/> Answer Self Test Review Questions 12-16.	_____
<input type="checkbox"/> Perform Experiment 1.	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Review the Examination Answers.	_____

THE BIPOLAR TRANSISTOR SWITCH

The basic component used in implementing any digital logic circuit is a switch. Modern digital integrated circuits use a high speed transistor switch as the primary component. There are two basic types of transistor switches used in implementing digital integrated circuits, the bipolar transistor and the metal oxide semiconductor field effect transistor (MOSFET). An understanding of these two devices is pertinent to the operation, capabilities, and limitations of the various types of digital integrated circuits.

The primary function of a transistor switch in a digital logic circuit is to alternately connect and disconnect a load to and from the circuit power supply. In doing this the transistor switch produces two distinct voltage levels across the load which represent the binary 0 and binary 1 states. The transistor switch should make and break these connections as quickly and efficiently as possible.

The most commonly used digital switch is the bipolar transistor. In digital applications, the bipolar transistor operates as an off/on or two state device. In one state the transistor is non-conducting or cut-off and acts as essentially an open circuit. In the other state, the transistor is conducting heavily and acts as a very low resistance, approaching a short circuit. A two state logic input signal is applied to the transistor to produce this on/off operation.

Modes of Operation

A bipolar transistor has three basic regions or modes of operation: cut-off, linear or active, and saturation. All three of these modes are used in digital circuits, the cut-off and linear mode in non-saturated bipolar circuits and the cut-off and saturation modes in saturated bipolar circuits.

Cut-Off. In the cut-off mode the transistor is nonconducting. Both the emitter-base (E-B) and collector-base (C-B) junctions are reverse biased or not biased at all to produce the cut-off state. In theory, no emitter or collector current flows, and the transistor acts as an open circuit between emitter and collector. In most practical transistors, however, the cut-off is not perfect. Because of imperfections in the semiconductor material out of which the device is made, some leakage current flows. In most modern transistors this leakage current is extremely low and for most practical applications can be neglected. However, where very high temperature operation is expected the leakage current becomes a more important consideration. In silicon transistors, the leakage current nearly doubles for each 10°C rise in temperature.

Linear. The linear or active mode of operation is characterized by a forward biased emitter-base junction and a reverse biased collector-base junction. In this mode, the transistor does conduct. Emitter and collector currents flow. The emitter and collector currents are directly proportional to the base current variations. The emitter and collector currents are simply amplified versions of the base current variation. In this mode of operation, the transistor functions as a variable resistance and is used to amplify or otherwise process analog signals. This region is of concern in non-saturating digital integrated circuits.

Saturation. The third mode of bipolar transistor operation is saturation. In this mode, both the emitter-base and collector-base junctions are forward biased. The transistor conducts heavily and acts as a very low resistance. The resistance between the emitter and collector is very low, approaching that of a short circuit.

In digital applications, the bipolar transistor is usually switched between the cut-off and saturation states. As it switches, the transistor passes quickly through the linear region. The primary responsibility of the designer of a digital circuit is to see that the bipolar transistor switches as quickly as possible between cut-off and saturation and that these two states are as stable as possible. In non-saturating digital circuits, the transistors switch between cut-off and the linear region. Again, high speed is the primary requirement.

Saturated Switching Circuits

Figure 2-1 shows the most common form of saturated bipolar transistor logic switch. Here, the transistor is connected as a shunt switch since it is in parallel with the output load R_L . This circuit is also known as a transistor logic inverter.

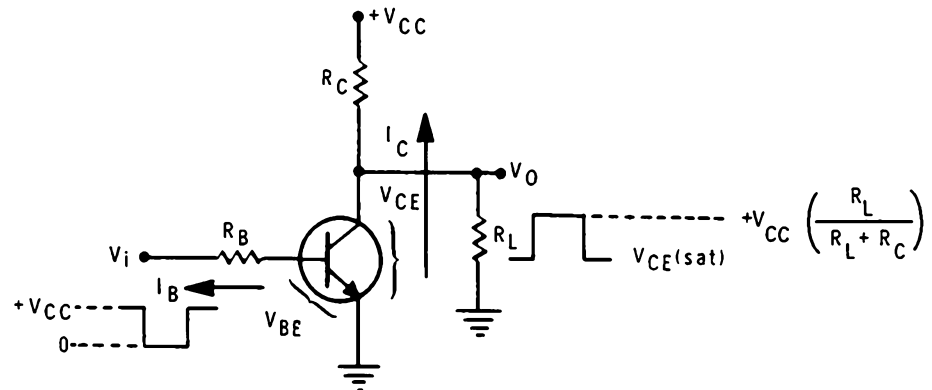


Figure 2-1

Basic transistor switching circuit.

With an input voltage V_i of zero volts or ground, the transistor is cut-off. The emitter-base junction is not forward biased, therefore, the transistor does not conduct. The only collector current flowing at this time is a minute leakage current. The output voltage V_O , with no load, is equal to the supply voltage V_{CC} as seen through collector resistor R_C . If a finite resistive load R_L is connected between the output and ground V_O will be some value less than V_{CC} and dependent upon the division ratio between R_C and R_L .

$$V_O = V_{CC} \left(\frac{R_L}{R_C + R_L} \right)$$

When an input voltage V_i of sufficient amplitude is applied to base resistor R_B , the emitter-base junction will become forward biased and the transistor will conduct. The transistor will be in the linear or saturation regions depending upon the size of V_i , the value of R_B and the gain (Beta or h_{FE}) of the transistor.

Figure 2-2 shows the typical input and output waveforms of a transistor switching circuit. The input switches between zero volts (LOW) and V_i (HIGH). When the input is LOW, the transistor is cut-off and output voltage V_o is approximately equal to V_{CC} (HIGH). When the input is HIGH, the transistor saturates and acts as a low resistance. The output voltage V_o is the collector-emitter saturation voltage $V_{CE}(\text{sat})$ which is only a few tenths of a volt (LOW). As you can see, the output is HIGH when the input is LOW, and the output is LOW when the input is HIGH. The input and output signals are always opposite one another. This is the reason for the name inverter.

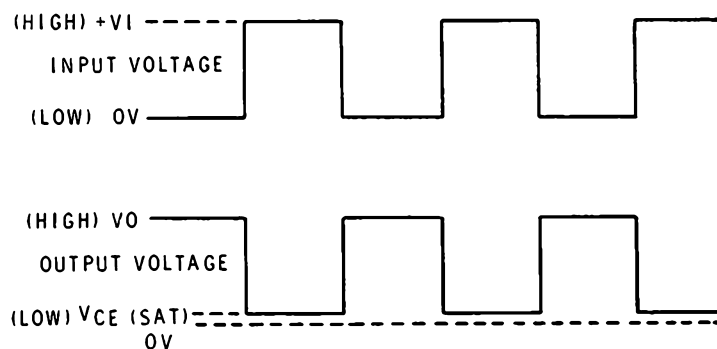


Figure 2-2

Input and output waveforms of a shunt transistor switching circuit.

In a digital circuit the magnitude of V_i and R_B are such that sufficient base current flows in order to cause the transistor to saturate. This condition occurs when the actual base current I_B is greater than the ratio of the collector current I_C to the dc forward current transfer ratio h_{FE} (also known as the dc current gain or Beta).

$$I_B > \frac{I_C}{h_{FE}} \text{ (for saturation)}$$

The base current is a direct function of the applied input voltage V_i and the value of the base resistor R_B . This relationship is

$$I_B = \frac{V_i - V_{BE}}{R_B}$$

where V_{BE} is the voltage across the forward biased emitter-base junction.

The ratio of the collector current (I_C) to the base current (I_B) in a common emitter circuit is known as the dc forward current transfer ratio or h_{FE} . This ratio

$$h_{FE} = \frac{I_C}{I_B}$$

expresses the effective gain of the device or the ability of the base current to control the larger collector current. The greater this ratio, the higher the gain.

By algebraically rearranging this expression, we can calculate the base current for a given collector current and gain.

$$I_B = \frac{I_C}{h_{FE}}$$

If we design the circuit so that the base current is greater than this ratio, the transistor will saturate. That is, both emitter-base and collector-base junctions will become forward biased. In this state the transistor is conducting heavily and the resistance between the emitter and collector is very low. For typical switching transistors this value of resistance is in the 5 to 30 ohms range. The voltage drop between the collector and emitter during saturation $V_{CE}(\text{sat})$ is only several tenths of a volt. This is very low compared to the supply voltage V_{CC} and therefore for most practical purposes is considered to be nearly zero volts.

During saturation the amount of emitter and collector current flowing becomes basically a function of the value of the supply voltage V_{CC} and the collector resistance R_C . Because the voltage drop across the transistor is essentially zero, the collector current can be found from the expression

$$I_C = \frac{V_{CC} - V_{CE}(\text{sat})}{R_C} \approx \frac{V_{CC}}{R_C}$$

This relationship holds true only if sufficient base current flows to saturate the transistor. If the value of the base current is less than the ratio of I_C/h_{FE} , the transistor will be operating in the linear region. The emitter-base junction will be forward biased but the collector base junction will be reverse biased. The collector-emitter voltage V_{CE} will be correspondingly higher.

There are two ways you can determine whether a transistor is saturated or operating in the linear region. The first method is to measure the junction potentials on the transistor. When measuring these voltages it is important to note the polarity of each transistor element with respect to the other. By knowing the magnitudes of the junction voltages and their relative polarities you can establish the state of the transistor.

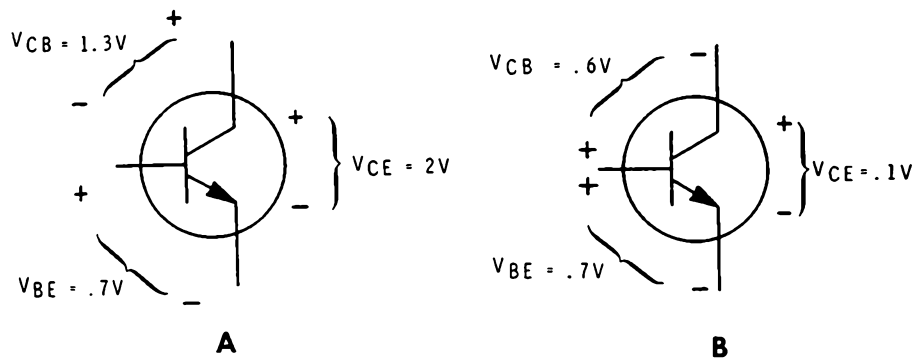


Figure 2-3

Polarity and voltage relationships in a conducting transistor (A) linear operation and (B) saturation.

Figure 2-3 shows typical junction voltages for an NPN transistor both in the saturated and unsaturated states. The transistor in Figure 2-3A is operating in the linear or active region. The emitter-base junction is forward biased because junction voltage V_{BE} is of the proper polarity. A conducting junction in a silicon transistor typically has a voltage drop of approximately .7 volt. Observing the collector-base junction we see that the voltage across it (V_{CB}) is 1.3 volts with the collector being more positive than the base. This indicates a reverse biased condition on the collector-base junction. The voltage drop between the emitter and collector V_{CE} is two volts. Note the relationship between V_{CE} , V_{BE} , and V_{CB} .

$$V_{CE} = V_{BE} + V_{CB}$$

In Figure 2-3B, the transistor is saturated. Again V_{BE} is approximately .7 volt with the polarity indicated. The big difference however is the polarity change in V_{CB} . Here the base is more positive than the collector indicating a forward biased condition on this junction. The junction voltage drop is approximately .6 volt. Again note that the collector-emitter voltage V_{CE} is the *algebraic* sum of (or difference between) V_{CB} and V_{BE} and in this case is only $.7 - .6 = .1$ volt. Because of this low voltage drop the effective resistance of the transistor is extremely low. Another beneficial characteristic of saturated operation is that the low collector-emitter voltage greatly minimizes the power dissipation in the conducting transistor. The power dissipation is

$$P = V_{CE} \cdot I_C$$

With V_{CE} so low during saturation, the power dissipation is also very low even though the collector current may be large.

The other method of determining whether or not a transistor is in saturation is to find both the base and collector current through actual circuit measurements and then determine if the base current is less than or greater than the ratio of the collector current and the dc current gain h_{FE} . As indicated before, if $I_B > I_C/h_{FE}$ then the transistor is saturated. If $I_B < I_C/h_{FE}$, the transistor is operating in its active region. The actual value of h_{FE} depends upon the type of transistor being used, the current level in the transistor, its temperature and other factors. The current gain value varies even among transistors of the same type. To ensure saturation in a switching transistor, designers generally make the base resistor small enough to produce a base current with a given minimum logic input voltage that is greater than the I_C/h_{FE} ratio. This safety factor is necessary in order to ensure saturation under all conditions. When the base current is greater than the I_C/h_{FE} ratio, we say that we are overdriving the base.

Self Test Review

1. The voltage drop across a conducting PN junction in a silicon transistor is approximately _____ volts.
2. The collector and base currents in a transistor inverter are measured and found to be $I_C = 10\text{ma}$, $I_B = .5\text{ ma}$. The transistor h_{FE} is 15. Is the transistor saturated? _____
3. A PNP transistor has the following junction voltages $V_{BE} = .7$ volts, base negative with respect to emitter. $V_{CB} = .5$ volts, collector positive with respect to base. Is the transistor saturated? _____
4. When a transistor is saturated it acts as a(n)
 - a. very low resistance
 - b. very high resistance
 - c. open circuit
 - d. variable resistance
5. To act as an open circuit, a transistor must be operating in which mode?
 - a. linear
 - b. saturation
 - c. cut-off
6. Another name for a saturated shunt transistor switch is _____ .

Answers


1. .7
2. No. The transistor is not saturated since
$$I_B < I_C/h_{FE}$$
$$I_B = I_C/h_{FE} = 10/15 = .666 \text{ ma}$$
Actual $I_B = .5 \text{ ma}$.
 I_B must be greater than I_C/h_{FE} for saturation.
3. Yes, the transistor is saturated. Both junctions are forward biased.
4. (a) Very low resistance
5. (c) Cut off
6. inverter



Switching Speed


One of the most important characteristics of a logic circuit is its ability to switch rapidly between the binary logic levels. This switching speed is affected by the transistor characteristics, the circuit component values, stray capacitance and inductance, the current and voltage levels in the circuit and the specific circuit configuration. When the input signal to a digital circuit changes from one logic level to the other, the output of the circuit does not change instantaneously. Instead, there is a delay time existing between the change in the input signal and the corresponding change in the output. This time lag is generally referred to as propagation delay.

The turn-on time of a transistor is primarily a function of the transistor characteristics and the amount of base drive applied to the circuit. A heavy base current helps to ensure a rapid turn-on.



The turn-off time delay is affected mainly by the transistor characteristics. The turn-off of the transistor is delayed because of storage time. When a transistor is saturated, an excess of minority carriers (holes in an NPN and electrons in a PNP transistor) build up in the collector-base junction region. This charge storage keeps the transistor conducting even with the base drive removed. It takes a finite period of time for this charge to be removed so that the transistor begins to come out of saturation. This storage time is a function of the transistor characteristics and the amount of base drive.

By using the proper transistors, circuit configuration, and component values, the switching delay times can be significantly reduced. Despite all of the factors that limit the switching speed, modern high-speed switching transistors can change from one state to the next in only a matter of nanoseconds (10^{-9} seconds).



Non-Saturating Switching Circuits

The most serious limitation to the switching speed of a bipolar transistor is the storage time associated with the condition of saturation. The charge storage build up in the collector-base region during saturation takes a finite time to be cleared away in order to turn the transistor off. This storage time is the most significant part of the turn off time and therefore any means of minimizing it will greatly increase the switching speed of the transistor. Special bipolar switching transistors have been designed to help minimize this storage time effect and external circuitry can be adjusted to minimize it to some degree. The obvious way to increase switching speeds, therefore, is to avoid saturation. By keeping the transistor from going into saturation no charge storage occurs therefore very high switching speeds can be achieved. A number of logic circuits have been designed using non-saturating bipolar transistor switches. The transistors switch between the cut-off and linear regions. Such circuits are capable of switching at frequencies as high as 1 GHz.

Self Test Review

7. The finite time that it takes transistor logic switches to turn on and off is called _____.
8. The turn on delay is a function of transistor characteristics and _____.
9. The turn off delay is caused primarily by _____.
10. Non-saturating logic circuits have a faster switching speed than saturated circuits.
 - a. True
 - b. False

Answers

- 7. propagation delay
- 8. base drive
- 9. charge storage
- 10. a. True

DESIGNING A SATURATED SWITCH LOGIC INVERTER

While most modern digital equipment is implemented with integrated circuits, it is sometimes necessary or desirable to supplement the IC with a discrete component circuit to perform a special function. This includes things such as logic level conversion, driving an indicator light such as an incandescent lamp or light emitting diode (LED) or operating a relay. All of these circuits use a saturated bipolar transistor switch. Several examples are given in Figure 2-4.

The circuit in Figure 2-4A is simply an inverter where the input and output logic levels may or may not be equal. Such a circuit is useful in interfacing different types of logic circuits. In Figure 2-4B, the LED will light when an input voltage V_i is applied. The transistor acts as an on/off switch controlled by V_i . This is also true in the relay driver circuit of Figure 2-4C. Contacts A and B of relay K1 are normally open (N.O.) when V_i is zero. When V_i becomes sufficiently positive, the transistor saturates and K1 is energized. The magnetic field produced by the relay coil closes contacts A and B. Diode D_1 is used to protect the transistor when the input voltage is removed. When the transistor cuts off the magnetic field the coil collapses thereby inducing a high negative voltage spike that can damage the transistor. The voltage spike causes D_1 to conduct and clamp the collector voltage to a safe level.

Such circuits are so common that it is desirable to know how to design them. The procedure is simple as you will see here.

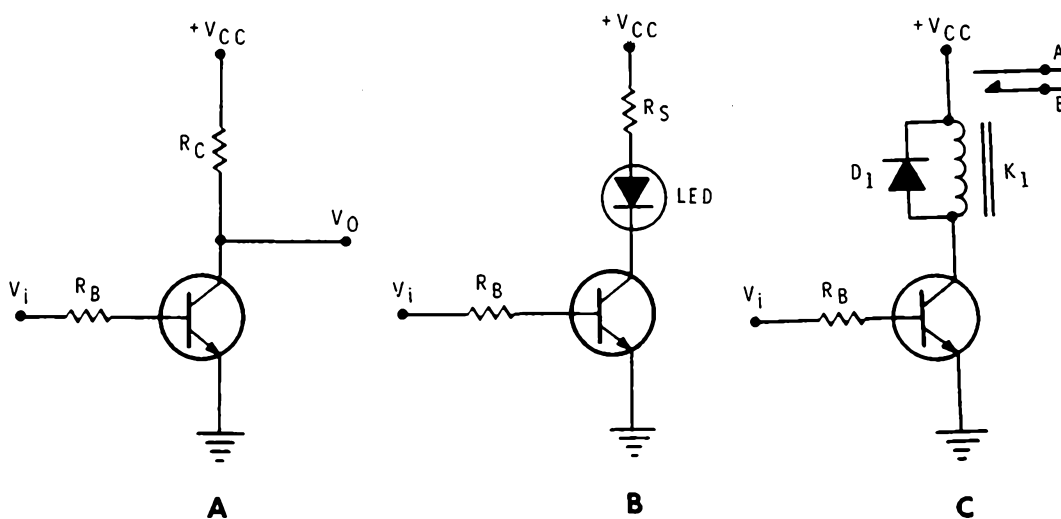


Figure 2-4

Bipolar transistor switch
used as a logic inverter (A),
an LED indicator driver (B),
and a relay drive (C).

Procedure

1. Define the load.

For a lamp or relay driver, the characteristics of the lamp or relay are given. These usually include a rated voltage (V_L), current and/or resistance. If the circuit is an inverter used to supply a signal to an external load, the load voltage and current or load resistance must be given.

2. Specify a supply voltage.

The supply voltage (V_{CC}) will usually be equal to or higher than the desired load voltage. Normally, the circuit will operate from an existing supply of some standard value such as +5 volts, +12 volts, etc.

3. Select a suitable transistor.

A wide variety of types are available. The exact application will guide you in its selection. A transistor designed for switching rather than linear applications is usually preferred. The voltage and current ratings will be determined by the load and supply characteristics. The collector current and voltage breakdown ratings should be at least twice the operating characteristics. Gain (h_{FE}) and operating speed requirements will be dictated by the application. The exact transistor characteristics can be determined from the manufacturer's data sheets.

4. Determine the value of any series dropping resistor.

If the supply voltage is larger than the load (lamp, relay, etc.) voltage, then some series dropping resistor (R_S) will be needed. See Figure 2-4B. The voltage across this resistance will be the supply voltage (V_{CC}) less the load voltage (V_L) and the transistor saturation voltage $V_{CE}(\text{sat})$. The resistance can then be found with the expression

$$R_S = \frac{V_{CC} - V_L - V_{CE}(\text{sat})}{I_C}$$

where I_C is the load current and the collector current $V_{CE}(\text{sat})$ is very low and can be considered zero. Use the closest standard resistor value. This step does not apply to an inverter with only a collector resistor.

5. Specify an output voltage.

If the circuit being designed is an inverter with a shunt load, the desired output voltage (V_o) should be specified. It will be some value less than V_{CC} depending upon the values of R_C and R_L . See Figure 2-5.

$$V_o = V_{CC} \left(\frac{R_L}{R_C + R_L} \right)$$

If there is no shunt load, $V_o = V_{CC}$.

6. Determine the value of the collector resistor.

If the circuit being designed is an inverter with a shunt load to ground, the collector resistor R_C value can be calculated if the supply voltage (V_{CC}), load resistance (R_L) and desired output voltage (V_o) are known. Since

$$V_o = V_{CC} \left(\frac{R_L}{R_C + R_L} \right)$$

The collector resistance can be found by rearranging this expression.

$$R_C = \frac{R_L (V_{CC} - V_o)}{V_o}$$

Use the closest standard value of resistance.

7. Determine the collector current.

This is the load current specified earlier for lamp or relay drivers. For the inverter arrangement described in Step 6, the current is

$$I_C = \frac{V_{CC} - V_{CE}(\text{sat})}{R_C} \approx \frac{V_{CC}}{R_C}$$

Since $V_{CE}(\text{sat})$ is usually only several tenths of a volt it can be considered negligible or zero in the above expression.

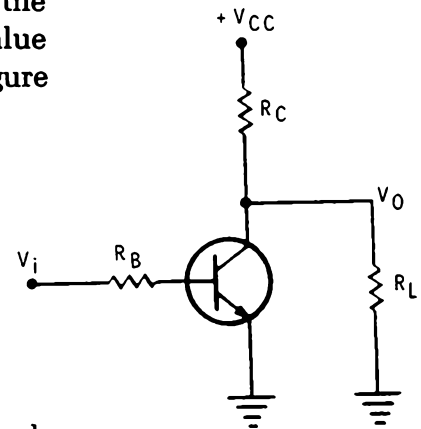


Figure 2-5

Logic inverter with shunt load.

8. Calculate the base current.

Knowing the gain (h_{FE}) from the manufacturer's data sheet and the collector current (I_C), the base current (I_B) can be found.

$$h_{FE} = \frac{I_C}{I_B} \text{ therefore } I_B = I_C/h_{FE}$$

Use the minimum value of h_{FE} quoted in the data sheet. To provide some overdrive and ensure saturation it is desirable to provide some safety factor by derating the h_{FE} by a factor of 2 (or more if desired). Therefore

$$I_B = \frac{I_C}{h_{FE}/2} \quad \text{or} \quad I_B = \frac{2 I_C}{h_{FE} (\text{min})}$$

9. Calculate the base resistor.

The base resistor (R_B) is found with the expression below.

$$R_B = \frac{V_i - V_{BE}}{I_B}$$

Here V_i is the lowest expected value of base drive voltage. It is the binary 1 voltage level for positive logic. V_{BE} of course is the emitter-base voltage drop which is typically .7 volts for a silicon transistor.

If the circuit being designed is to be driven by another inverter as in Figure 2-6 its supply voltage and collector resistance should be considered.

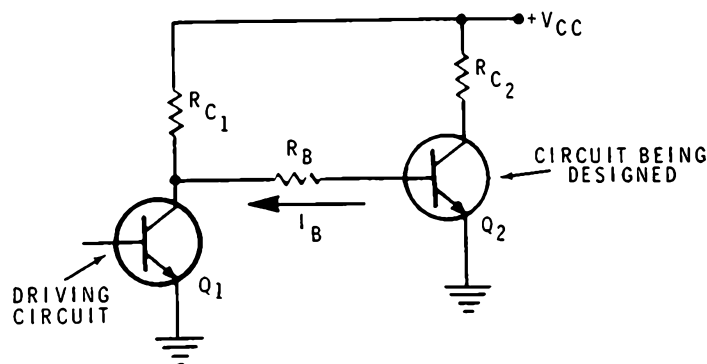


Figure 2-6

When Q_1 cuts off, the driver circuit will be activated by I_B flowing through R_B and the collector resistor R_{C1} of the driving circuit. In this case,

$$R_B = \frac{V_{CC} - I_B R_{C1} - V_{BE}}{I_B}$$

Use the closest lower standard resistance value.

The following examples will illustrate the procedure for two practical applications.

Example Application 1

Design a driver for an LED indicator. The steps below correspond to those in the procedure. See Figure 2-4B.

1. The load is a light emitting diode indicator whose normal brilliance is obtained with 20 ma of current. Its normal operating voltage, the load voltage drop (V_L), is 1.7 volts at this current value.
2. The supply voltage V_{CC} is + 5 volts.
3. A type MPSA20 silicon transistor will be used. $h_{FE}(\text{min}) = 100$.
4. Since the LED voltage drop is less than the supply voltage a series dropping resistor is needed. The load (collector) current is 20 ma.

$$R_S = \frac{V_{CC} - V_L - V_{CE}(\text{sat})}{I_C}$$

If we consider $V_{CE}(\text{sat})$ negligible then

$$R_S = \frac{5 - 1.7}{.02} = \frac{3.3}{.02} = 165 \text{ ohms}$$

A standard 150 or 180 ohm 10 percent resistor or 160 ohm 5 percent resistor could be used.

5. Not applicable.

6. Not applicable.

7. $I_C = 20 \text{ ma}$ or .02 amp.

$$8. I_B = \frac{2 I_C}{h_{FE}} = \frac{2 (.02)}{100} = \frac{.04}{100} = .4 \text{ ma} = 400 \mu\text{A}$$

9. Assume the input voltage $V_i = +3.5 \text{ volts}$. $V_{BE} = .7 \text{ volts}$ for a silicon transistor.

$$R_B = \frac{V_i - V_{BE}}{I_B} = \frac{3.5 - .7}{.0004} = 7000 \text{ ohms}$$

A standard 6.8K 10 percent value can be used.

See Figure 2-7

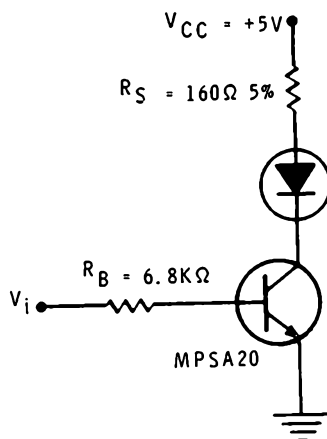


Figure 2-7

LED driver circuit:
Example Application 1.

Example Application 2

Design an inverter circuit to apply +6 volts across a 600 ohm load. The supply voltage is +15 volts. The driving signal (V_i) is supplied by the +15 volt supply through a 1000 ohm collector resistor. See Figure 2-8.

1. $R_L = 600$ ohms, $V_L = 6$ volts.
2. $V_{CC} = +15$ volts.
3. Use an MPSA20.
4. Not applicable.
5. $V_o = 6$ volts.

$$R_C = \frac{R_L(V_{CC} - V_o)}{V_o}$$

$$R_{C2} = \frac{600(15 - 6)}{6} = \frac{600(9)}{6} = 900 \text{ ohms}$$

A standard 910 ohm resistor could be used.

7. $I_C = \frac{V_{CC}}{R_C} = \frac{15}{910} = .0165 \text{ or } 16.5 \text{ ma}$
 8. $I_B = \frac{2 I_C}{h_{FE}} = \frac{2(.0165)}{100} = .00033 \text{ amp} = .33 \text{ ma} = 330 \mu\text{A}$
 9. $R_B = \frac{V_{CC} - I_B R_{C1} - V_{BE}}{I_B}$
- $$R_B = \frac{15 - (.00033)(1000) - .7}{.00033} = \frac{13.97}{.00033} = 42,333 \text{ ohms}$$

A standard 39K ohm resistor could be used. See Figure 2-8.

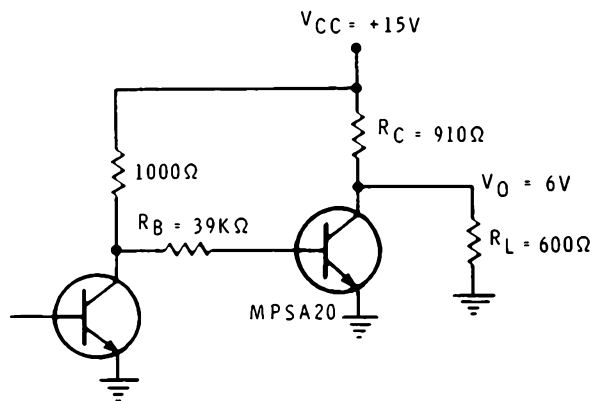


Figure 2-8

Inverter circuit:
Example Application 2.

Self Test Review

11. Design a transistor switch circuit that will energize a relay whose coil resistance is 400 ohms and current requirement is 30 ma. Use a supply voltage of +15 volts. The input voltage is +15 volts. Use an MPSA20 transistor. Draw the completed circuit and label all values.

Answers

11. The steps below correspond to those in the procedure.

1. $R_L = 400 \text{ ohms}$, $I_L = I_C = 30 \text{ ma}$, $V_L = R_L I_C = 400(.03) = 12 \text{ volts}$

2. $V_{CC} = +15 \text{ volts}$

3. MPSA20 transistor. $h_{FE}(\text{min}) = 100$

4. $R_S = \frac{V_{CC} - V_L - V_{CE}}{I_C} = \frac{15 - 12}{.03} = \frac{3}{.03} = 100 \text{ ohms}$

5. Not applicable.

6. Not applicable.

7. $I_C = I_L = 30 \text{ ma}$

8. $I_B = \frac{2I_C}{h_{FE}(\text{min})} = \frac{2(.03)}{100} = \frac{.06}{100} = .0006 \text{ ma} = 600 \mu\text{A}$

9. $R_B = \frac{V_i - V_{BE}}{I_B} = \frac{15 - .7}{.0006} = \frac{14.3}{.0006} = 23,833$

Use a standard 22 K ohm resistor.

See Figure 2-9.

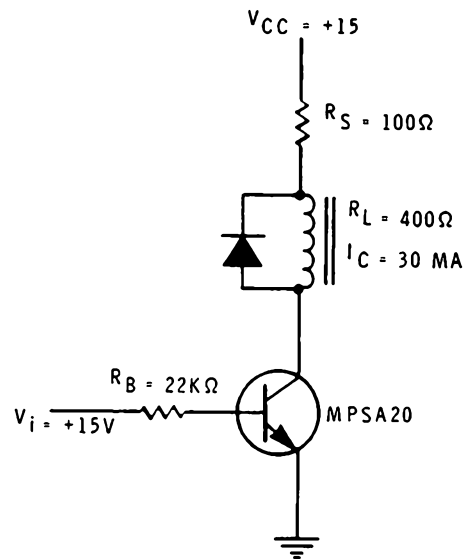


Figure 2-9

Solution to Self Test
Review problem 11.

MOS FIELD EFFECT TRANSISTORS

Another transistor widely used in digital integrated circuits is the enhancement mode metal oxide semiconductor field effect transistor (MOSFET). Also known as an insulated gate field effect transistor (IGFET), this device offers numerous advantages over the bipolar transistor particularly for digital work and integrated circuits. Because the MOSFET is basically a simple device, it is more easily constructed than a bipolar transistor. Since it can be made smaller than bipolar devices it permits more circuitry to be produced on a given area of semiconductor material. The cost of an integrated circuit is directly proportional to the area of the semiconductor material used to construct the circuit and the complexity of the devices used. Simple, low cost, high density circuits are readily constructed with MOSFET components.

The N-channel MOSFET

Figure 2-10 shows the basic construction of an N-channel enhancement mode MOSFET. It is constructed of a P-type silicon base or substrate into which is diffused two N-type semiconductor material areas. These form the source and drain elements of the transistor. On top of this is diffused a thin layer of silicon dioxide, a glass insulator which isolates the source and drain regions from the remainder of the device. On top of the silicon dioxide insulator is formed a third element called the gate. The silicon dioxide insulates the gate terminal from the P-type silicon material. The gate is simply a metallized diffusion such as aluminum or a silicon conductive material that forms a capacitor with the P-type silicon base, the silicon dioxide layer acting as the dielectric. The area beneath the gate dielectric and between the source and drain is known as the channel. If the drain is made positive with respect to the source, current will flow between the source and the drain. It is the level and polarity of the voltage between the source and the gate that determines the conductivity of the channel.

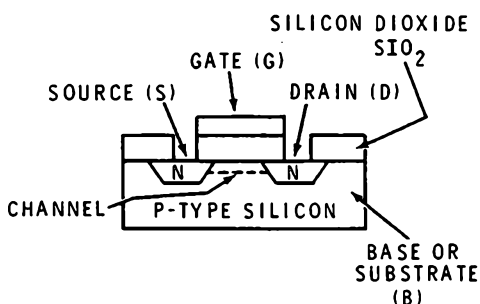


Figure 2-10

N-channel enhancement
mode MOSFET

With the gate source voltage equal to zero, no current flows between the source and the drain. The alternate N-type and P-type materials between source and drain effectively form two back-to-back diodes both of which are cut off. However, when the voltage applied between the gate and source exceeds a certain threshold level with the gate positive with respect to the source, an electric field will be established in the channel region. This causes the transistor to conduct and electrons will flow between the source and the drain.

The gate-source arrangement in the MOSFET acts as a capacitor. Applying a gate-source voltage charges this capacitor. The gate becomes positive and the area below the gate in the substrate becomes negative. The majority carriers in the P-type substrate (holes) will be depleted by the negative charge and the electron density will be enhanced. This negative charge in the P-type base establishes a channel for current flow between the two N-type regions. Removing the gate-source voltage or decreasing it below the threshold level will cause the conduction to cease.

The enhancement mode MOSFET is an excellent switch. When the gate voltage is below the threshold value, the resistance between the source and the drain is extremely high and very closely approximates an open circuit. When the gate voltage is above the threshold level, the transistor conducts and the resistance between the source and the drain is very low, approximating a short circuit. Such characteristics make the enhancement mode MOSFET ideal for digital circuits.

One of the primary benefits of the MOSFET over the standard bipolar transistor is that the input impedance between source and gate is extremely high. The input impedance between the source and gate is many thousands of megohms and, in effect, is more capacitive than resistive. This high input impedance minimizes the loading of one logic circuit on the next. Figure 2-11 shows the schematic symbol normally used to represent an N-channel enhancement mode MOSFET.

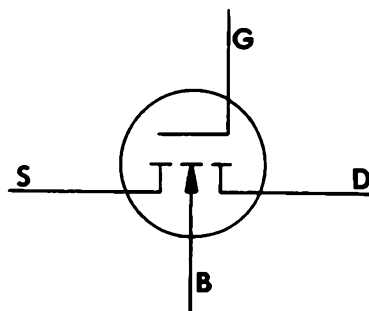


Figure 2-11

Schematic symbol for an N-channel enhancement mode MOSFET.

The P-Channel MOSFET

Figure 2-12 shows the construction of a P-channel enhancement mode MOSFET. It is similar in construction to the N-Channel device. However, the substrate is N-type material, while the source and drain diffusions are P-type material. The symbol used to represent this device is shown in Figure 2-13. The operation of the P-channel MOSFET is similar to that for the N-channel device with the exception of the operating voltage polarities. Like the N-channel device, the P-channel MOSFET is normally off if the gate-to-source voltage is below a specific threshold value. When the gate is made negative with respect to the source and is of an amplitude above the threshold level, the transistor will conduct and current will flow between the source and the drain. In the P-channel device, the current carriers are holes instead of electrons.

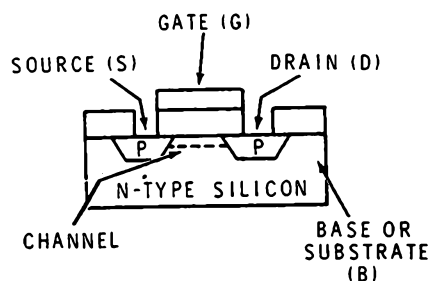


Figure 2-12

P-Channel enhancement
mode MOSFET

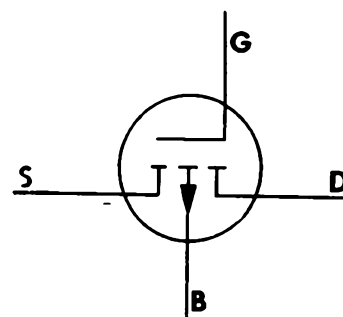


Figure 2-13

Schematic symbol of a
P-channel enhancement
mode MOSFET.

One of the most important characteristics of the enhancement mode MOSFET is its threshold voltage, that value of gate-source voltage required to cause the transistor to conduct. There are two basic types of threshold in common use; low threshold and high threshold. The high threshold devices are easier to make, and, therefore, are more common. This high threshold value is approximately three to four volts. There is circuitry available with a low threshold in the one to two volt region. Typically, P-channel devices have high thresholds, while N-channel devices have low thresholds.

Figure 2-14 shows a simplified schematic symbol often used to represent an enhancement mode MOSFET. Because of the complexity of the standard symbols, they are rarely used. The letter P or N is written adjacent to the symbol to designate whether it is a P-channel or an N-channel device.

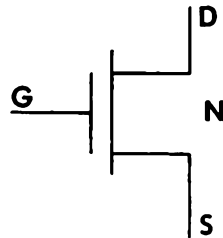


Figure 2-14

Simplified symbol of
enhancement mode MOSFET.

Bipolars vs. MOSFETs

The primary benefits of the MOSFET over the bipolar transistor are; small size, simplicity of construction, high input impedance and low power consumption. However, these advantages are somewhat offset by the major disadvantage; slow switching speeds. Because of the high impedance nature of the device and its capacitive characteristics, switching speeds are significantly lower than those for bipolar transistors. Recent advances have improved the switching speed of MOSFETs, but switching speeds in the region below 100 nanoseconds are not possible. Nevertheless, the advantages of the MOSFET over the bipolar for many applications offset this switching speed disadvantage. Bipolar transistors are still faster and offer the further advantage of being able to handle high power applications.

MOSFET Circuits

Logic circuits are readily constructed with enhancement mode MOSFETs. Figure 2-15 shows an inverter circuit constructed with a P-channel MOSFET. With the gate input voltage (V_i) at zero, the transistor does not conduct and the output voltage (V_o) is the drain supply voltage (V_{DD}) as seen through the drain resistor (R_D). When the input voltage is made sufficiently negative beyond the threshold value, the transistor conducts and acts virtually as a short circuit. The output voltage at this time is near zero volts. An inverter using an N-channel device would be similar, but with positive logic levels and supply voltage.

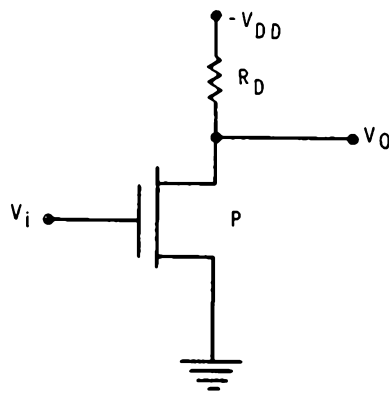


Figure 2-15

P-channel MOSFET inverter

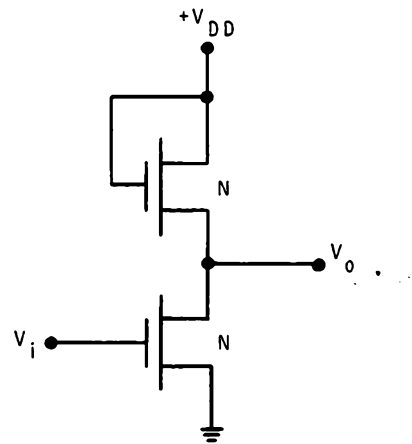


Figure 2-16

Using a biased MOSFET
as a load resistor.

In practical MOS circuits, the drain resistor (R_D) is not used. Because it occupies a substantial amount of space in an integrated circuit, it is generally eliminated and replaced by another MOSFET, biased to act as a resistance. The transistor itself is smaller than a diffused resistor. This is illustrated in Figure 2-16.

A very popular type of MOS digital circuit combines both P-channel and N-channel devices to form what is known as a complementary MOS (CMOS) logic circuit. You will learn more about these devices in a later unit.

Self Test Review

12. The MOSFET is also known as a _____ .
13. To cause an N-channel enhancement mode MOSFET to conduct the gate-source voltage must be
 - a. zero
 - b. negative
 - c. positive
 - d. above the threshold level
14. The gate-source appears to external circuits primarily as a
 - a. short circuit
 - b. low impedance
 - c. capacitor
 - d. high impedance
15. The switching time of a MOSFET compared to that of a bipolar transistor is
 - a. less
 - b. more
 - c. about the same
16. To cause a P-channel MOSFET to conduct, the gate-source voltage must be higher than the _____ and the drain must be _____ with respect to the source.

Answers

- 12. IGFET
- 13. (c) positive, gate positive with respect to the source and
(d) above the threshold value
- 14. (c) capacitor (d) high impedance
- 15. (b) more
- 16. threshold, negative

EXPERIMENT 1

Bipolar Transistor Switch

OBJECTIVES:

To demonstrate the operation, characteristics and design of a saturated bipolar transistor switch.

Materials Needed:

- 1 – NPN transistor MPSA20 (417-801)
- 1 – $560\ \Omega$ resistor
- 1 – $1\ \text{k}\Omega$ resistor
- 1 – $4.7\ \text{k}\Omega$ resistor
- 1 – $220\ \text{k}\Omega$ resistor

Voltmeter (vom or DMM)

Heathkit Digital Design Experimenter (Refer to the ET-3200 manual for operational details and breadboarding procedures.)

Procedure

1. Construct the circuit shown in Figure 2-17. The circuit receives its input from data switch SW1. You will monitor the output at the collector with a voltmeter. See Figure 2-18 for transistor base details.

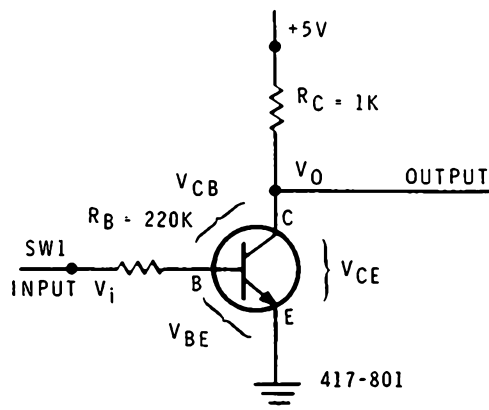


Figure 2-17

Inverter circuit for Step 1.

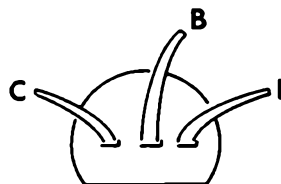


Figure 2-18

Lead connections
for 417-801 transistor.

2. Set SW1 to the down (LO) position and measure the dc output voltage (V_O) with respect to ground. Record below.

$V_O =$ _____ volts.

2. The input voltage (V_i) at this time is

$V_i =$ _____ volts.

3. Set SW1 to the up (HI) position and measure the circuit dc input voltage (V_i) with respect to ground. Record below.

$V_i =$ _____ volts.

Next, measure the following transistor junction voltages base-to-emitter (V_{BE}), base-to-collector (V_{CB}), and collector-to-emitter (V_{CE}). Record your values in Table I. Be sure to note the polarity of each voltage so that you can determine whether the junctions are forward or reverse biased. Study your results, then answer the following question. Is the transistor saturated? _____

TABLE I

R_B	V_{BE}	V_{CB}	$V_{CE} = V_O$
220 K			
4.7 K			

4. Measure the voltage drop across the 220K base resistor (V_{RB}) and the voltage across the collector resistor (V_{RC}) and record in Table II. Using these voltages and associated resistor values, calculate the base current (I_B) and collector current (I_C) using Ohm's law.

$$I_B = \frac{V_{RB}}{R_B}$$

$$I_C = \frac{V_{RC}}{R_C}$$

Record your values in Table II. Also calculate the ratio I_C/I_B and record in Table II.

TABLE II

	$R_B = 220K$	$R_B = 4.7K$
V_{RB}		
V_{RC}		
I_B		
I_C		
I_C/I_B		

5. Using the criterion that states a transistor is saturated if the $I_C/I_B < h_{FE}$ and the data in Table II, determine the condition of the transistor assuming $h_{FE} = 100$.

Is the transistor saturated? _____

6. Replace the 220K ohm base resistor R_B with a 4.7k ohm resistor.
7. Repeat Step 3. Measure V_{BE} , V_{CB} , and V_{CE} and record in Table I. Study your results.

Is the transistor saturated? _____

8. Repeat Step 4 recording your data in Table II.

9. Repeat Step 5. Is the transistor saturated? _____

Discussion of Steps 1 Through 9.

In Step 1 you constructed a bipolar transistor switch. In Step 2 with the input from SW1 (LO or ground), the E-B junction is not forward biased, therefore the transistor is cut off and the output voltage you measured was $V_{CC} = +5$ volts as seen through the 1k collector resistor.

In Step 3 you applied forward bias to the emitter-base junction of the transistor from the +5 volt logic level output of SW1 through the 220K base resistor. The transistor conducts. This is indicated by the junction voltages. You should have measured a V_{BE} in the .6 to .8 volts range with the base positive with respect to the emitter. V_{CB} should have been one volt or so with the collector positive with respect to the base. The collector to emitter voltage V_{CE} should have been 1 to 4 volts with the collector the most positive element. The emitter-base junction is forward biased and the base-collector junction is reverse biased. See Figure 2-19 for a summary of the transistor junction voltage polarities for a properly biased transistor. With this arrangement, the transistor conducts but is not saturated. The transistor is operating somewhere in its linear region.

Because we do not know the exact value of h_{FE} , your voltages for V_{CB} and V_{CE} may not be exactly equal to those given above but they should be close.

In Steps 4 and 5 you determined the collector and base currents by measuring the voltage drops across the base and collector resistors then dividing by the respective resistor values. Then you determined if the ratio I_C/I_B was equal to, less than, or greater than the assumed h_{FE} value of 100. If $I_C/I_B < h_{FE}$ then the transistor is saturated. In this step you should have found I_C/I_B to be higher than h_{FE} . Obviously, the transistor is not

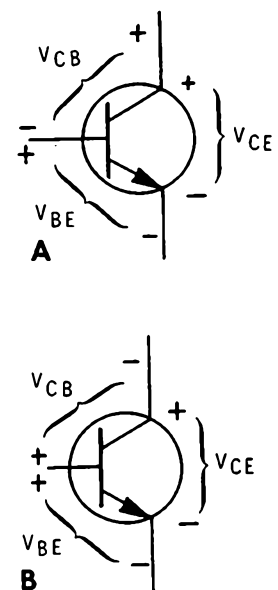


Figure 2-19

NPN Transistor
Junction bias polarities.

saturated. What you calculated when you divided I_C by I_B was the true gain or h_{FE} of the transistor. It should have been greater than 100.

In Step 6, you replaced the 220 k Ω base resistor with a 4.7 k Ω resistor. This provides more base drive current. In Step 7 with +5 volts applied to the input, the transistor conducts, but this time much harder. The junction voltages should be approximately as follows: $V_{BE} = .7$ volt, $V_{CB} = .6$ volt, $V_{CE} = .1$ volt. The polarities of V_{BE} and V_{CE} are as before. But with the 4.7 k Ω base resistor, V_{CB} is such that the base is more positive than the collector. This indicates that the base-collector junction is forward biased whereas, with the 220 k Ω base resistor, it was reverse biased. With both the emitter-base and base-collector junctions forward biased, the transistor is saturated. The output voltage V_O or V_{CE} is the difference between V_{BE} and V_{CB} . See Figure 2-19B.

In Steps 8 and 9, you again determined I_C , I_B and I_C/I_B . In this case, you should have found I_C/I_B to be less than a nominal h_{FE} of 100. Therefore, the transistor is saturated.

10. Using the procedure outlined earlier, design a transistor switch to produce +4 volts across a 560 ohm shunt load. Use a +12-volt supply voltage and a MPSA20 (417-801) transistor (h_{FE} min = 100). The input logic signal will come from data switch SW1 and is +5 volts as seen through a 470 Ω resistor. (See Schematic of ET 3200 Digital Design Experimenter.) Calculate all values. Draw the completed circuit. Then breadboard the circuit using the closest available resistor values and check its operation. Use one of the procedures described earlier to insure that the transistor does saturate.

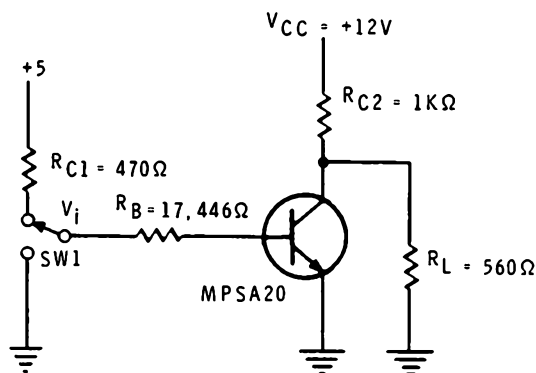


Figure 2-20

Solution to design problem in Step 10.

Discussion of Step 10

Your calculations should appear like those shown below. See Figure 2-20.

$$R_{C2} = \frac{R_L(V_{CC} - V_o)}{V_o} = \frac{560(12 - 4)}{4} = \frac{560(8)}{4}$$

$$R_{C2} = 1120 \text{ ohms}$$

Use a 1K ohm resistor

$$I_C = \frac{V_{CC}}{R_{C2}} = \frac{12}{1000} = .012 \text{ amp} = 12 \text{ ma}$$

$$I_B = \frac{2I_C}{h_{FE}} = \frac{2(.012)}{100} = \frac{.024}{100} = .00024 \text{ amp} = 240 \mu\text{A}$$

$$R_B = \frac{V_{CC} - I_B R_{C1} - V_{BE}}{I_B} = \frac{5 - (.00024)(470) - .7}{.00024}$$

$$R_B = \frac{4.1872}{.00024} = 17,446 \text{ ohms}$$

A standard 15 K or 16 K Ω resistor could be used. A 10 K Ω resistor in series with a 4.7 K Ω resistor (14.7 K Ω total) could be used in breadboarding this circuit on your ET-3200.

In the calculation for R_B , V_{CC} is 5 volts since the 5 volt supply in your ET-3200 provides the basic source of base drive through the 470 ohm resistor associated with logic switch SW1. This resistor is designated R_{C1} in Figure 2-20.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this Unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and answer all of the questions first before checking your answers.

1. A bipolar transistor acts as a closed switch when it is:
 - a. cut-off
 - b. saturated
 - c. amplifying
2. The transistor amplifies if it is operating in which of the following bias regions?
 - a. cut-off
 - b. saturation
 - c. linear
3. No current flows in a transistor when it is:
 - a. cut-off
 - b. saturated
 - c. amplifying
4. A transistor, when used in switching or in logic applications, acts like a(n):
 - a. amplifier
 - b. oscillator
 - c. variable resistor
 - d. switch
5. The circuit whose binary output is opposite of its input is called a(n):
 - a. amplifier
 - b. inverter
 - c. converter
 - d. switch
6. A transistor is biased such that its emitter-base junction is forward biased and its base-collector junction is reversed biased. The transistor is operating in which region?
 - a. linear
 - b. cut-off
 - c. saturation

7. When both emitter-base and base-collector junctions on a transistor are forward biased, the transistor is operating in which region?
- linear
 - cut-off
 - saturation
8. A transistor inverter with a supply voltage of +5 volts, a collector resistance of 1K ohms and the h_{FE} of 50 has a base current of .3 ma. The transistor is saturated.
- True
 - False
9. Refer to Figure 2-21. Identify each of the transistor symbols.

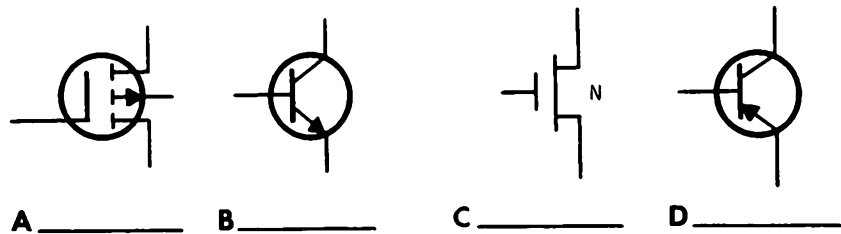


Figure 2-21

Figure for Question 9.

10. Refer to Figure 2-22. With the bias voltages shown, the transistor is operating in which region?
- linear
 - cut-off
 - saturation

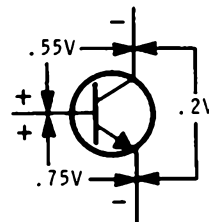


Figure 2-22

Illustration for
Question 10.

11. A P-channel MOSFET has a threshold voltage of -3.5 volts. The source to gate voltage is -7 volts. The transistor is:
 - a. conducting
 - b. cut-off
12. Which of the following is **not** a reason why MOSFETs are preferred over bipolar transistors in integrated circuits?
 - a. They are smaller.
 - b. They consume less power.
 - c. They are faster.
 - d. They are simpler.
13. An N-channel MOSFET with a threshold of $+1.5$ volts is not conducting. Which of the following source gate voltages is applied?
 - a. $+0.7$ volt
 - b. $+2$ volts
 - c. $+10$ volts
14. The factor that most influences the switching speed of a saturated bipolar transistor is:
 - a. collector current
 - b. base current
 - c. h_{FE}
 - d. charge storage
15. You could most improve the switching speed of a bipolar transistor by:
 - a. decreasing base current
 - b. increasing collector current
 - c. avoiding saturation
 - d. using a different transistor

EXAMINATION ANSWERS

1. b.—saturated
2. c.—linear
3. a.—cut-off
4. d.—switch
5. b.—inverter
6. a.—linear
7. c.—saturation
8. a.—True

$$I_C = \frac{+5}{1000} = .005\text{A or } 5\text{ma}$$

$$I_B = \frac{I_C}{h_{FE}} = \frac{5}{50} = .1\text{ma}$$

$$\text{For saturation, } I_B > \frac{I_C}{h_{FE}}$$

Since I_B is .3ma, the transistor is saturated.

9. a.—P-channel MOSFET
b.—NPN bipolar
c.—N-channel MOSFET
d.—PNP bipolar
10. c.—saturation. Both junctions are forward biased.
11. a.—conducting. The gate-source voltage, -7 volts, is higher than the threshold, -3.5 volts, therefore the MOSFET is conducting.
12. c.—They are faster. This is **not** a reason why MOSFETs are preferred over bipolars. Bipolars switch faster than MOSFETs.
13. a.— $+ .7$ volts. This value is below the threshold of $+1.5$ volts, therefore, the MOSFET is cut-off.
14. d.—charge storage
15. c.—avoiding saturation

Unit 3

**DIGITAL LOGIC
CIRCUITS**

CONTENTS

Introduction	3-3
Unit Objectives	3-4
Unit Activity Guide	3-5
Types of Logic Circuits	3-6
The Inverter	3-9
Decision-Making Logic Elements	3-12
The Dual Nature of Logic Gates	3-20
NAND/NOR Gates	3-22
Practical Logic Circuits	3-30
Experiment 2 — Logic Inverter	3-35
Experiment 3 — Diode Logic Gates	3-41
Experiment 4 — Transistor Logic Gate	3-46
Unit Examination	3-50
Examination Answers	3-54
Appendix — Positive and Negative Logic Equivalent Circuits	3-56

INTRODUCTION

All digital equipment, simple or complex, is constructed from just a few basic circuits. These circuits are called logic elements. A logic element performs some specific logic function on binary data.

There are two basic types of digital logic circuits: decision-making and memory. Decision making logic elements monitor binary inputs and produce outputs based on the input states and the operational characteristics of the logic element. Memory elements are used to store binary data.

Whether the digital equipment is a simple piece of test equipment or a large scale digital computer, this equipment is made up entirely of such logic circuits. Once you learn the basic logic elements and the most commonly used forms, you will be able to understand and determine the operation of any piece of digital equipment. By understanding how to apply these basic logic elements, you will be capable of designing and troubleshooting digital equipment.

In this unit you are going to study the basic types of digital logic circuits. You will learn the operation of decision-making and memory circuits. You will also discover several special classes of logic circuits. We will also discuss the practical implementation of these logic circuits using switches and relays, discrete electronic components and integrated circuits. This unit provides the base upon which all of the remaining units in this program are built. The study of digital techniques is, in effect, a study of digital logic circuits and how they are applied. As a result, this is an extremely important background lesson.

UNIT OBJECTIVES

When you complete Unit 3 you will have the following knowledge and capabilities.

1. You will be able to list the three basic types of logic elements.
2. You will be able to write a definition for combinational logic circuits.
3. You will be able to write a definition for sequential logic circuits.
4. You will be able to draw the schematic and explain the operation of both diode and switch contact AND gates.
5. You will be able to draw the schematic and explain the operation of both diode and switch contact OR gates.
6. You will be able to draw the schematic and explain the operation of a transistor and switch contact inverter.
7. Given a list of symbols, you will be able to identify the industry standard symbols for inverters, AND, OR, NAND, and NOR gates.
8. From a list of truth tables, you will be able to identify the logic functions being performed.
9. You will be able to write a truth table for any of the basic logic functions: AND, OR, NOT, NAND, NOR.
10. Given a list of logic equations you will be able to identify the logic function expressed by each.
11. You will be able to write the logic equation for any of the basic logic functions AND, OR, NAND, NOR, NOT.

UNIT ACTIVITY GUIDE

Completion
Time

- | | |
|---|-------|
| <input type="checkbox"/> Read "Types of Logic Circuits." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 1-6. | _____ |
| <input type="checkbox"/> Read "The Inverter." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions: 7-11. | _____ |
| <input type="checkbox"/> Read "Decision-Making Logic Elements." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 12-23. | _____ |
| <input type="checkbox"/> Read "NAND/NOR Gates." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 24-33. | _____ |
| <input type="checkbox"/> Read "Practical Logic Circuits." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 34-39. | _____ |
| <input type="checkbox"/> Perform Experiment 2. | _____ |
| <input type="checkbox"/> Perform Experiment 3. | _____ |
| <input type="checkbox"/> Perform Experiment 4. | _____ |
| <input type="checkbox"/> Complete the Unit Examination. | _____ |
| <input type="checkbox"/> Review the Examination Answers. | _____ |

TYPES OF LOGIC CIRCUITS



Figure 3-1

Basic symbol of a logic gate.

The two basic types of logic circuits are decision-making and memory. Both types accept binary inputs and produce binary outputs. The nature of the output is a function of the state of the inputs and the characteristics of the particular logic circuit.

Decision-making logic circuits do exactly what their name implies; make decisions. The basic decision-making logic element is called a gate. A gate has two or more binary inputs and a single output. Figure 3-1 shows the generalized symbol used for representing a logic gate. More specific symbols are used to represent practical types of gates. There are several different types of logic gates, each performing a specific decision-making operation. The gate looks at its binary inputs, and based upon their states and its operation, it generates an appropriate output signal that reflects the decision it has made.

While many simple logic functions can be implemented with a gate, generally gates are combined to form more sophisticated and complex decision making logic networks called combinational circuits. A combinational circuit is made up of two or more gates and has two or more inputs and one or more outputs. Combinational circuits still perform a decision-making function but of a more sophisticated nature. Most combinational circuits perform some unique logic function such as decoding, encoding, multiplexing, comparison or an arithmetic operation with binary numbers. A generalized block diagram of a combinational logic circuit is shown in Figure 3-2.

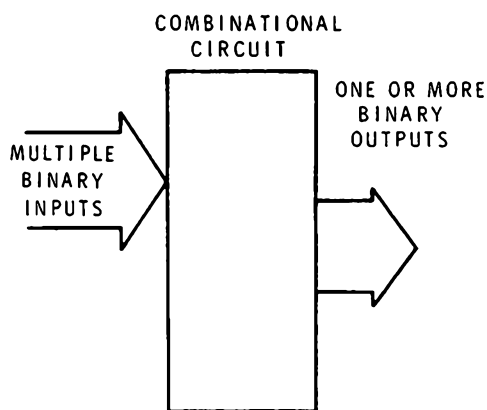


Figure 3-2

General block diagram of a combinational logic circuit.

The other type of logic element is a memory circuit. The basic memory element is a bistable storage device known as a flip-flop. This circuit has two stable states which can represent the two binary numbers 0 and 1. The circuit can be placed into either state so that it retains that state or remembers the bit stored there. Most memory circuits store a single binary bit. Many of these elements can be combined to store complete binary numbers or words.

Most memory elements are interconnected with combinational circuits to form another more sophisticated form of memory element known as a sequential circuit. A general block diagram of a sequential circuit is shown in Figure 3-3. The inputs to the sequential circuit consists of external binary data and feedback signals developed within the sequential circuit itself. The outputs of a sequential circuit are binary signals that are used to operate or control external circuits. The output of the sequential circuit is a function of the binary inputs, the binary data stored in the sequential circuit itself and the specific characteristics of this circuit.

Sequential circuits are used for a variety of operations in digital equipment. Typical sequential circuits consist of counters, shift registers, timers, sequencers, and other circuits where binary data is stored and manipulated as a function of time. You will study these circuit in later units.

The three basic decision-making logic elements are the AND gate, the OR gate, and the inverter. All other digital logic elements and circuits are variations or combinations of just these three basic elements. Each of these elements receives one or more binary inputs and generates a single binary output.

In order to distinguish one binary input signal from another and in order to identify both binary inputs and outputs, each signal is generally assigned a name or label. In its simplest form the label consists of a letter, a small word or mnemonic (pronounced ne-mon-ik). This label or designation indicates a specific logic signal which can assume either of the two binary states 0 or 1. In the discussions of the logic circuits presented here all inputs and outputs will be given such names and labels. In most cases a simple letter designation will be used. However, keep in mind that these designations may be words or letter and number combinations. Some typical examples are A, X, D3, CLR, JMPZ, etc.

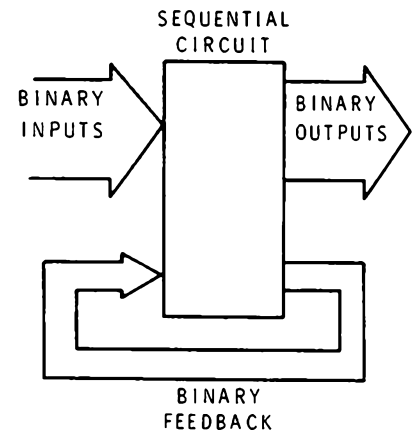


Figure 3-3

General block diagram of a sequential logic circuit.

Self Test Review

1. The two basic classifications of logic circuits are _____ and _____.
2. When multiple gates are interconnected to perform a specific function, the resulting circuit is called a _____ circuit.
3. The basic logic memory element is called a _____ and is capable of storing one _____ of data.
4. Combining memory elements with a combinational logic circuit produces a _____ logic circuit.
5. The three basic logic elements are the _____, _____, and _____.
6. A point in a logic circuit labelled STB3 can assume which of the following states:
 - a. binary 0.
 - b. binary 1.
 - c. either binary 1 or 0.

Answers

1. decision-making, memory
2. combinational
3. flip-flop, bit
4. sequential
5. AND, OR, inverter
6. (c) either binary 0 or 1.

THE INVERTER

The simplest form of a digital logic circuit is the inverter or NOT circuit. The inverter is a logic element whose output state is always opposite of its input state. If the input is a binary 0 the output is a binary 1. If the input is a binary 1, the output is a binary 0. We say that the inverter has an output that is the complement of the input. The binary states 1 and 0 are considered to be complementary.

The operation of the inverter is clearly summarized by a simple chart known as a truth table. The truth table shows all possible input states and the resulting outputs. Figure 3-4 is the truth table for an inverter. The input to the inverter is designated A while the output is labeled \bar{A} (pronounced A NOT or NOT A). The bar over the letter A indicates the complement of A . Note that the truth table shows all possible input combinations and the corresponding output for each. Since the inverter has a single input, there are only two possible input combinations: 0 and 1. The output in each case is the complement or opposite of the input.

INPUT	OUTPUT
A	\bar{A}
0	1
1	0

Figure 3-4

Truth Table for logic inverter.

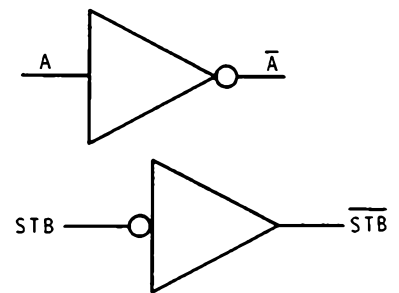


Figure 3-5

Symbols for a logic inverter.

The simplest and most widely used form of logic inverter is the transistor switching circuit shown in Figure 3-6. This inverter circuit operates with a binary input signal whose logic levels are 0 volts (or ground) and some positive voltage approximately equal to the supply voltage $+V_{CC}$. When the input B to the transistor is 0 volts, or ground, the emitter-base junction of the transistor is not forward biased. Therefore, the transistor does not conduct. With the transistor cut-off, the output, \bar{B} , is at the supply voltage potential $+V_{CC}$. As you can see, with positive logic level assignments, the binary 0 input produces a binary 1 output.

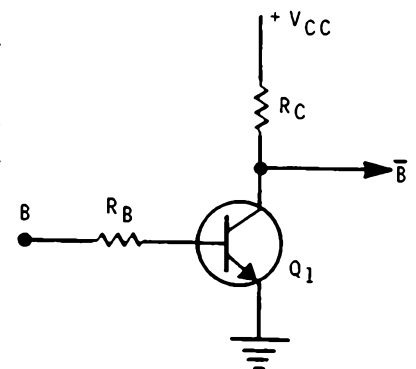


Figure 3-6

A transistor logic inverter.

Whenever a binary 1 or positive voltage level approximately equal to $+V_{CC}$ is applied to the input, the emitter-base junction of the transistor becomes forward biased. Sufficient base current flows through the circuit to cause the transistor to saturate. When the transistor is saturated, both the emitter-base and base-collector junctions of the transistor are forward biased and it acts as a very low impedance. At this time, the complement output \bar{B} is approximately equal to zero volts or ground as seen through the conducting transistor. The actual voltage will be equal to the saturation voltage of the transistor $V_{CE(sat)}$. For most good high speed switching transistors this voltage is several tenths of a volt or less and for most practical purposes can be considered to be zero. With a positive voltage binary 1 input the output is a binary 0. This circuit obviously performs logic inversion. The input and output voltage waveforms representing the operation of this circuit are shown in Figure 3-7. Most modern switching transistors are capable of turning off and on at nanosecond speeds, therefore very high frequency operation with this circuit is possible.

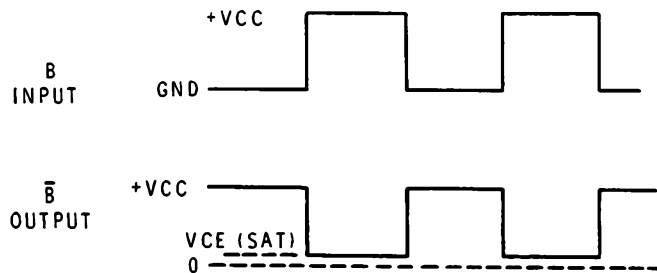


Figure 3-7
Input and output waveforms
of a transistor logic inverter.

Self Test Review

7. If the input to a logic inverter is labeled PLS, the output will be
 - a. PLS
 - b. SLP
 - c. $\overline{\text{PLS}}$
 - d. binary 0
8. The inverter input PLS is a voltage level that represents
 - a. binary 0
 - b. binary 1
 - c. either binary 0 or binary 1
 - d. Cannot be determined with information given.

9. If the input to the inverter in Figure 3-6 is simply left open the transistor will not conduct. What will the output be if negative logic level assignments are assumed?
 - a. binary 0
 - b. binary 1
 - c. Cannot be determined with information given.
10. Saturation means
 - a. both emitter-base and base-collector junctions reverse biased.
 - b. both emitter-base and base-collector junctions forward biased.
 - c. emitter-base junction forward biased, base-collector junction reverse biased.
 - d. emitter-base junction reverse biased, base-collector junction forward biased.
11. The output of the inverter 2 in Figure 3-8 with a binary 1 input will be
 - a. binary 0
 - b. binary 1
 - c. Cannot be determined with information given.

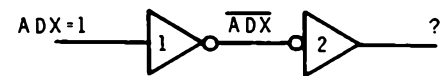


Figure 3-8

Answers

7. (c) $\overline{\text{PLS}}$ The output of an inverter is indicated by the input designation with a NOT bar over it to indicate the complement.
8. (c) either binary 0 or binary 1. The input designation PLS represents a logic variable that can assume either state.
9. (a) binary 0. If the input to the transistor logic inverter is left open, the transistor will not conduct. Therefore, the output will be $+V_{cc}$. With negative logic level assignments $+V_{cc}$ = binary 0 and zero volts or ground = binary 1.
10. (b) both emitter-base and base-collector junctions forward biased. Saturation means that the transistor conducts hard and represents a very low impedance. Both transistor junctions are forward biased.
11. (b) Binary 1. If the input ADX is binary 1 the output of inverter 1 $\overline{\text{ADX}}$ will be binary 0. With a binary 0 input to inverter 2, its output will be the complement or binary 1. The input ADX and output are the same. The output could also be called ADX. In this circuit, one inverter cancels the effect of the other. The output will be equal to the input for any *even* number of cascaded inverters.

DECISION-MAKING LOGIC ELEMENTS

The two basic types of decision-making logic elements are the AND gate and the OR gate. These are logic circuits with two or more inputs and a single output. The output state is a function of the input states and how the particular gate operates. The gate makes its decision based upon the input states and its particular function, then generates the appropriate binary output. Let's consider each of these basic gates in detail.

The AND Gate

The AND gate is a logic circuit that has two or more inputs and a single output. The operation of the gate is such that the output of the gate is a binary 1 if and only if all inputs are binary 1. If any one or more inputs are a binary 0 the output will be binary 0. The AND gate is a control circuit whose output is a binary 1 only when all inputs to the gate are in the binary 1 state at the same time.

INPUTS		OUTPUT
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Figure 3-9

Truth Table for an AND gate.



Figure 3-10

Logic Symbol for AND gate.

The operation of a two input AND gate is indicated by the truth table in Figure 3-9. The inputs are designated A and B. The output is designated C. The output for all possible input combinations is indicated in the truth table. The total number of input combinations is determined by raising 2 to a power equal to the number of inputs. With two inputs, each capable of assuming either of the two binary states, the total possible number of combination inputs is $2^2 = 4$. Note that the output is binary 0 for any set of inputs where either, or both, of the inputs are binary 0. The output is binary 1 only when both inputs are binary 1.

The basic symbol used to represent an AND gate is shown in Figure 3-10. Note that the inputs and outputs are labeled to correspond to the truth table in Figure 3-9. Keep in mind that the AND gate may have any number of logical inputs.

An important point to note about the AND symbol in Figure 3-10 is the equation at the output, $C = A \cdot B$ or $C = AB$. This equation is a form of algebraic expression that is used to designate the logical function being performed. The equation expresses the output C in terms of the input variables A and B and is read "C equals A AND B". Here the AND function is designated by the dot between the two input variables A and B. The AND function is designated by an expression similar to the product of algebraic variables.

As you will see, the operation of all logic elements and gates can be expressed in the form of an algebraic equation. These expressions permit circuits to be analyzed, designed and optimized by using standard algebraic operations and special algebraic manipulations designated by rules of Boolean algebra. Boolean algebra is a special form of two state algebra that is useful in working with binary variables. You will learn more about this in a later unit.

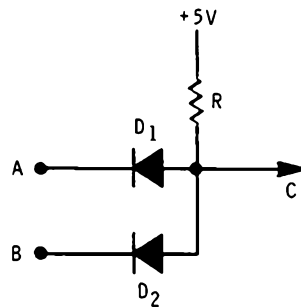


Figure 3-11

Two input diode AND gate.

The circuit in Figure 3-11 shows one electrical implementation of a logic AND gate. Here two diodes and a resistor form an electronic circuit that produces the AND function using binary signals. To analyze this circuit, assume the use of binary input signals using positive logic designations of zero volts (ground) and +5 volts. Let's also assume that perfect diodes are used. Keep in mind that in a practical circuit real diodes have a threshold voltage level before they conduct and a finite voltage drop across them during conduction.

Now let's analyze the operation of the AND circuit shown in Figure 3-11. If both inputs are a binary 0 (zero volts or ground) both diodes conduct. Assuming no forward voltage drop across the diode, the output will be zero volts. If either one of the inputs is a binary 0, while the other is at the binary 1 level (+5 volts), the diode associated with the input whose state is binary 0 will be forward biased and will conduct thereby clamping the output at the binary 0 level. (The diode associated with the input at the binary 1 level will be reverse biased and cut off.)

If both inputs are at the binary 1 level, neither diode will conduct and the output will be +5 volts. As you can see, the only time the output is at the binary 1 level is when both inputs are binary 1.

The operation of the AND gate shown in Figure 3-11 is fully illustrated by the voltage truth table in Figure 3-12 and the waveform timing diagram in Figure 3-13. Note that if positive logic is assumed, the voltage truth table corresponds to the logic truth table given in Figure 3-9. In the wave form diagram, both inputs A and B are shown switching at various times between the binary 0 and binary 1 levels of ground and +5 volts. The output C corresponding to this particular combination of inputs is also illustrated. Note that the only time that the output is binary 1 is when the inputs are both in the binary 1 state. The output is a binary 1 for a period of time during which the two inputs are coincidentally at the binary 1 level. The AND gate is sometimes referred to as a coincident gate.

INPUTS		OUTPUT
A	B	C
0V	0V	0V
0V	+5V	0V
+5V	0V	0V
+5V	+5V	+5V

Figure 3-12

Voltage truth table for diode AND gate.

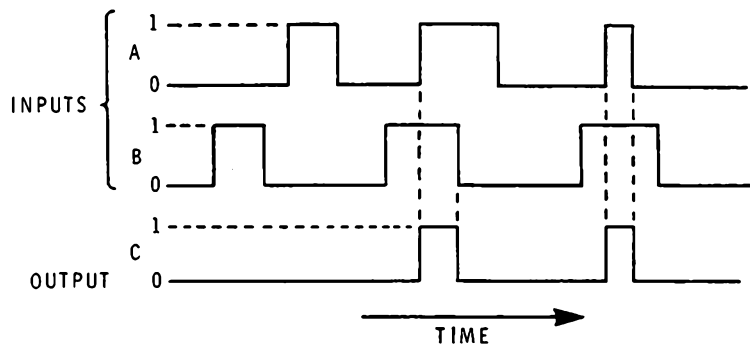


Figure 3-13

Input and output waveforms of the diode AND gate.

Figure 3-14 shows one of the most common applications of the AND gate in digital circuits. Here one input of the AND gate is used to control the passage of the other input signal to the output. The input control signal CTL enables or inhibits the passage of the other logic signal which is a train of square waves designated SQW. The output is identical to SQW during the time the CTL input is binary 1. Note the input and output waveforms as well as the logic expression for this function indicated in Figure 3-14.

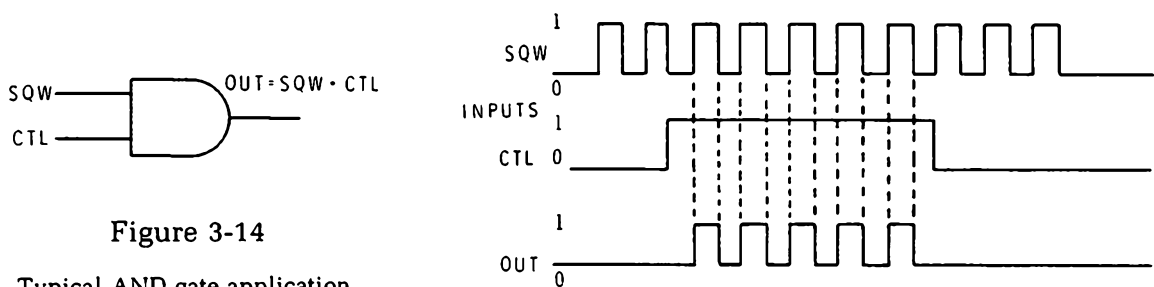


Figure 3-14

Typical AND gate application.

Remember that an AND gate may have more than two inputs, the exact number being dictated by the application. In addition, there are many other ways of implementing the logical AND function with hardware. Later in this unit and in the program you will learn about some of these circuits.

Self Test Review

12. Draw the logic symbol for an AND gate with inputs J7, K6, L4, and output F3.
13. Write the logic equation for an AND gate with inputs XLT, ZMO, KMD, A3 and output TF. _____
14. The logic gate in Figure 3-15 will have how many different input combinations?
 - a. 4
 - b. 8
 - c. 16
 - d. 32
15. Write the truth table for a three-input AND gate with inputs A, B, C, and D as the output.
16. What is the algebraic output equation for the circuit in Figure 3-16?
 - a. $P = \overline{MN}$
 - b. $P = M\overline{N}$
 - c. $P = \overline{M}N$
 - d. $P = MN$

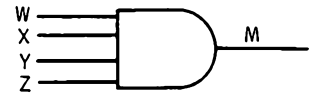


Figure 3-15

Circuit for Self Test Review Question 14.

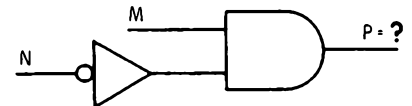


Figure 3-16

Circuit for Self Test Review Question 16.

Answers

12. See Figure 3-17
13. $TF = XLT \cdot ZMO \cdot KMD \cdot A3$ The AND function is designated by the dot between each variable.
14. (c) 16 The four input gate in Figure 3-15 can have a total of $2^4 = 16$ different input combinations.



Figure 3-17

Answer to Self Test Review Question 12.

15.

INPUTS			OUTPUT
A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Figure 3-18

Truth Table for 3-input AND gate.

All possible input combinations are indicated. Note that the input states correspond to the 3-bit binary numbers representing the decimal numbers 0 through 7. This is a convenient way of listing all input state variations for a given number of input bits.

16. (b) $P = M\bar{N}$. The output of the inverter in Figure 3-16 is \bar{N} . This is then ANDed with input M and the output expression is constructed by writing the input variables adjacent to one another.

The OR Gate

INPUTS		OUTPUT
D	E	F
0	0	0
0	1	1
1	0	1
1	1	1

Figure 3-19

Truth Table for OR gate.

The other basic logic element is called an OR gate. Like the AND gate it can have two or more inputs and a single output. Its operation is such that the output is a binary 1 if any one or all inputs are a binary 1. The output is binary 0 only when both inputs are binary 0.

The logical operation of a two input OR gate is expressed by the truth table in Figure 3-19. With two inputs there are $2^2 = 4$ possible input combinations as explained earlier. The truth table designates all four possible input combinations and the corresponding output. Note that the output is binary 1 when either or both of the inputs are binary 1. The output is binary 1 if the D input, OR the E input, OR both are present.

The logical symbol for an OR gate is shown in Figure 3-20. The inputs are labeled according to the truth table in Figure 3-19. Note the output algebraic expression for the OR gate $F = D + E$. The plus sign is used to designate the logic OR function. The output F is expressed in terms of the input logic variables D and E .

The circuit in Figure 3-21 illustrates the implementation of the logic OR function using semiconductor diodes. This gate is similar to the AND circuit considered earlier except that the supply voltage and diode polarities are reversed. This circuit operates with logic input levels of zero and +5 volts as did the AND gate considered previously. Using positive logic designations, let's evaluate the operation of this circuit.

When both inputs D and E are at binary 0 (zero volts or ground) both diodes conduct. Assuming perfect diodes, with no forward voltage drop, the output will be binary 0. If either one of the logical inputs is a binary 0, while the other is a binary 1 (+5 volts), the diode associated with the input at the binary 1 state will conduct causing the output to be +5 volts or a binary 1. The diode associated with the input at the binary 0 state will be reversed biased and will not affect the circuit. When both inputs are at the binary 1 or +5 volt level, both diodes conduct and the output is a +5 volt or binary 1 level. The voltage truth table for this circuit is shown in Figure 3-22. Using positive logic, it corresponds to the table in Figure 3-19.

The operation of the OR gate is illustrated more fully by the waveforms shown in Figure 3-23. These diagrams show the output state for various combinations of the input voltage D and E as a function of time. Note that if one of the inputs switches to the binary 1 level the output switches to binary 1. The output is binary 1 when either or both inputs are binary 1. Study the waveforms in Figure 3-23 carefully at each point to be sure that you understand the OR function.



Figure 3-20
Logic symbol for OR gate.

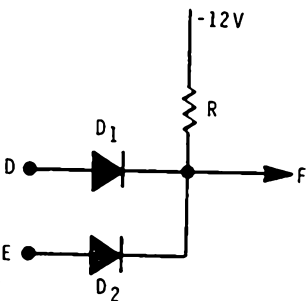


Figure 3-21
Diode OR gate.

INPUTS		OUTPUT
D	E	F
0V	0V	0V
0V	+5V	+5V
+5V	0V	+5V
+5V	+5V	+5V

Figure 3-22
Voltage Truth Table
for the diode OR gate.

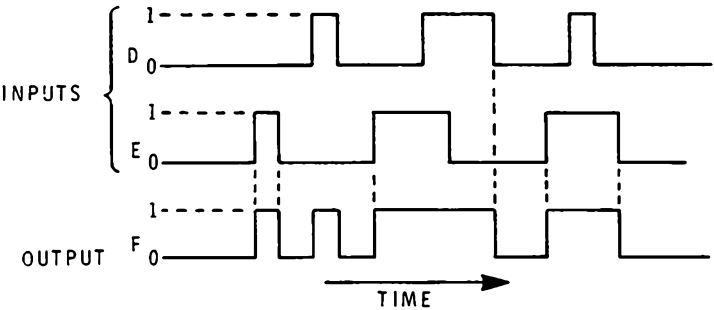


Figure 3-23
Input and output waveforms
of the diode OR gate.

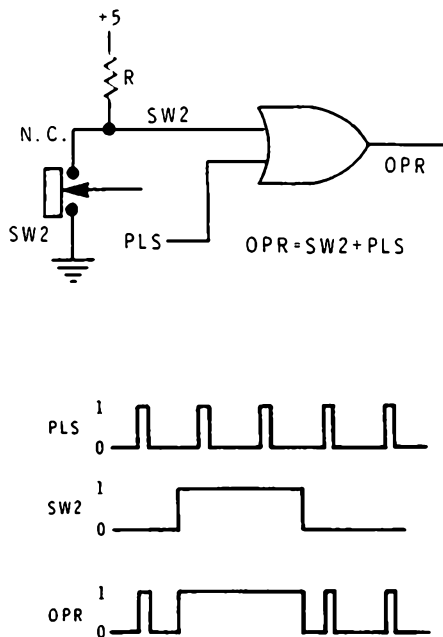


Figure 3-24

Typical OR gate application.

Figure 3-24 illustrates a typical application for the OR gate in a digital circuit. There are two inputs to the OR gate in this application: a push button switch SW2 and a train of pulses designated PLS. The output OPR will be binary 1 when either SW2 or PLS is at the binary 1 (+5 volts) level. Switch SW2 is a normally closed pushbutton. The SW2 input to the OR gate is normally ground or binary 0. When the switch is depressed, the contacts open and the SW2 input becomes +5 volts or binary 1 level as seen through the resistor. The other input PLS is a series of pulse trains that switch momentarily between the binary 0 and binary 1 levels. Note the algebraic expression for the output, $OPR = SW2 + PLS$. The accompanying waveforms in Figure 3-24 illustrate the operation of the circuit. The OR gate permits either of the two inputs to control the output.

When considering the operation and application of an OR gate you should remember that this logic element can have two or more inputs as called for by the application. At the same time we have only illustrated one method of implementing the OR gate with electronic hardware. Many other circuit variations are used and these will be discussed later in the Program.

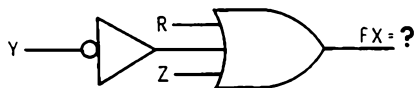


Figure 3-25

Circuit for Self Test Review Question 18.

Self Test Review

17. Draw the logic symbol for an OR gate with inputs GB, PH, CD, SH, and output FF.
18. Write the output equation of the gate in Figure 3-25.
19. Write the truth table for a 4-input OR gate with inputs W, X, Y, Z, and output J.
20. The output of an OR gate is binary 0 when:
 - a. all inputs are binary 0.
 - b. any one or more inputs are binary 0.
 - c. all inputs are binary 1.
 - d. any one or all inputs are binary 1.
21. The logic OR function when expressed in algebraic terms is analogous to the:
 - a. product
 - b. sum
 - c. difference
 - d. quotient

Answers

17. See Figure 3-26. Note that the + signs between the two-letter input variables designate the OR operation.
18. $FX = R + \bar{Y} + Z$. The inverter complements the Y input before it is ORed with the other variables.
- 19.

INPUTS				OUTPUT
W	X	Y	Z	J
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Figure 3-27

Truth Table for 4-input OR gate

For four input variables there are $2^4 = 16$ possible input combinations. These can be determined by simply writing the four bit numbers 0 through 15 as indicated. Note that the output is a binary 1, if any one or more inputs is binary 1.

20. (a) All inputs are binary 0. The only time the output of an OR gate is binary 0 is when all inputs are binary zero. At all other times at least one or more inputs is binary 1, thereby producing a binary 1 output.
21. (b) sum. The ORing of logic inputs is analogous to the summing of algebraic variables. The AND function is analogous to the product of input variables.

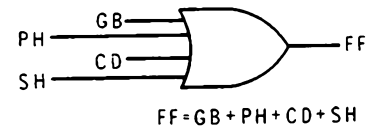


Figure 3-26

Answer to Self Test
Review Question 17.

THE DUAL NATURE OF LOGIC GATES

When we explained the operation of the diode gate circuit in Figure 3-11, we indicated that it performs the logic AND function. We proved this by considering the output voltage level for each of the four possible combinations of input voltage levels. The voltage truth table for this gate is repeated here in Figure 3-28A. In considering the operation of this circuit, we assumed the use of positive logic level assignments. By doing this we were able to translate the voltage levels given in the truth table into the table shown in Figure 3-28B. Naturally this table clearly indicates that the AND function is being performed. The output is a binary 1 only when both inputs are binary 1.

INPUTS		OUTPUT
A	B	C
0V	0V	0V
0V	+5V	0V
+5V	0V	0V
+5V	+5V	+5V

A

INPUTS		OUTPUT
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

B

INPUTS		OUTPUT
A	B	C
1	1	1
1	0	1
0	1	1
0	0	0

C

Figure 3-28

Truth Table for diode gate in
Figure 3-11.

Now let's consider the function of this circuit when we assume negative logic level assignments. In this case the 0 volt level would represent a binary 1 and a +5 volt level would represent a binary 0. Using the original data as developed in Figure 3-28A and translating it into a truth table using binary 1's and 0's with negative logic level assignments, we obtain the truth table shown in Figure 3-28C. By studying this truth table, you will see that the circuit no longer appears to be performing the AND function. Close inspection of the truth table will reveal that the circuit is now performing the OR function since the output is a binary 1 when either one or both of the inputs is a binary 1. (The order or sequence of the inputs is not the same as that for the AND gate, but this is not important. It is the function that counts.) Our only conclusion can be that with positive logic the circuit in Figure 3-11 performs AND function but with negative logic the circuit performs the OR function. This clearly indicates that the diode gate circuit is capable of performing either of the two basic logical functions. Refer to the Appendix on Page 3-56 for comparison of positive and negative logic equivalent circuits.

This dual nature of logic gates applies to any logic circuit. The diode gate that you considered in Figure 3-21 is also dual in nature. With positive logic level assignments it performs the OR function as indicated previously. However, if you analyze the circuit by using negative logic level assignments you will find that the circuit performs the AND function. Keep this important fact in mind as it will help you in analyzing, troubleshooting and designing digital circuits. You must not only know how the circuit operates electrically, but also what logic level assignments are being used.

Figure 3-29 shows the logic symbols normally used to represent gates that perform the logic AND and OR functions with negative logic level assignments. The circles at the inputs and outputs represent the effect of reversing the logic level assignments from positive to negative.

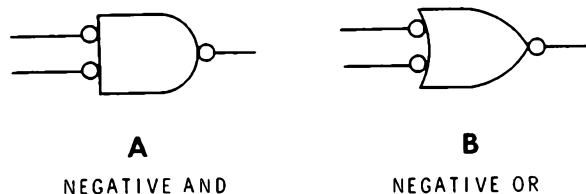


Figure 3-29

Negative logic gates

Self Test Review

22. Any logic circuit can perform both AND and OR operations.
 - a. True
 - b. False
23. A given logic circuit performs the AND function when binary 0 = 0 volts and binary 1 = -6 volts. Reversing the logic level assignments makes the gate perform as a _____ gate.

Answers

22. (a) True
23. Positive OR

NAND/NOR GATES

While many digital circuits can be constructed with just the three basic digital logic elements – AND, OR, and NOT – most digital equipment is implemented with special versions of these circuits known as NAND and NOR gates. Such circuits are basically AND and OR gates combined with an inverter. NAND/NOR gates are the most widely used types of digital logic elements because they offer numerous advantages over the simple diode gates considered earlier. In large complex digital logic networks, it is difficult to cascade more than just a few of the simple diode logic gates. Because there is no buffering between the gates, loading problems occur and the speed of operation suffers. For that reason, it is generally desirable to combine a simple diode logic gate with some type of transistor buffer to permit more flexible interconnection of circuits. This transistor buffer is most often an inverter.

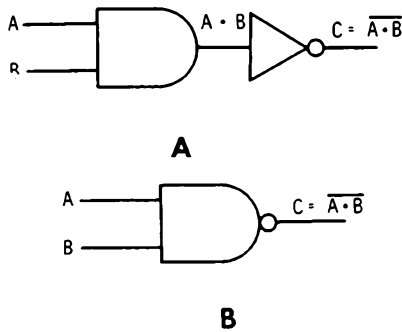


Figure 3-30

NAND gate.

INPUTS		OUTPUT	
A	B	AND $A \cdot B$	NAND $A \cdot B = C$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Figure 3-31

Truth Table of NAND gate.

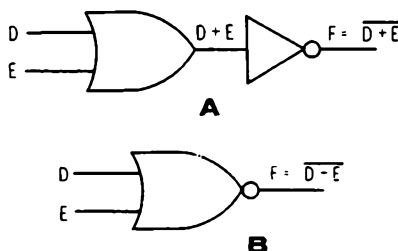


Figure 3-32

NOR gate.

NAND Gate

The term NAND is a contraction of the expression NOT-AND. A NAND gate, therefore, is an AND gate followed by an inverter. Figure 3-30A shows the basic diagram of a NAND gate. Note the algebraic output expression for the AND gate and the inverter. The entire AND output expression is inverted and indicated by the bar over it.

Figure 3-30B shows the standard symbol used for a NAND gate. It is similar to the AND symbol, but a circle has been added at the output to represent the inversion that takes place.

The logic operation of the NAND gate is easy to infer from the circuit in Figure 3-30. This operation is indicated by the truth table in Figure 3-31. The NAND output is simply the complement of the AND output.

NOR Gate

Like the NAND gate, the NOR gate is an improved logic element used for implementing decision-making logic functions. The term NOR is a contraction for the expression NOT-OR. Therefore, the NOR gate is essentially a circuit combining the logic functions of an OR gate and an inverter.

Figure 3-32A is a logical representation of a NOR gate. Figure 3-32B shows the standard symbol used to represent a NOR gate. Note that the output expression is the inverted OR function.

The logic operation of a NOR gate is illustrated by the truth table in Figure 3-33. The NOR output is simply the complement of the OR function. Like any other logic gate, NAND and NOR gates may have two or more inputs as required by the application.

NAND and NOR gates can be used to implement any of three basic logic functions. For example, by tying all inputs together, either the NAND or NOR gate performs inversion. By combining the NAND or NOR gates with external inverters, the AND and OR operations can be performed.

INPUTS		OUTPUT	
D	E	OR $D + E$	NOR $\overline{D + E} = F$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Figure 3-33

Truth Table of NOR gate.

Self Test Review

24. NAND/NOR gates are more widely used than simple AND/OR circuits because:
 - a. NAND/NORs can also perform AND/OR operations.
 - b. NAND/NORs are less expensive and smaller.
 - c. NAND/NORs are self-buffering – this permits higher speed and reasonable loading.
 - d. AND/ORs can't perform the NOT function.
25. A 3-input NAND gate has inputs of 0, 1 and 1. The output is:
 - a. binary 0
 - b. binary 1
26. A 4-input NOR gate has inputs of 1, 0, 0 and 0. The output is:
 - a. binary 0
 - b. binary 1
27. The output equation of a NAND is:
 - a. $C = A \cdot B$
 - b. $C = A + B$
 - c. $C = \overline{A \cdot B}$
 - d. $C = \overline{A + B}$
28. The output equation for a NOR is:
 - a. $F = D + E$
 - b. $F = D \cdot E$
 - c. $F = \overline{D \cdot E}$
 - d. $F = \overline{D + E}$

Answers

- 24. (c) NAND/NORs are self-buffering – this permits higher speeds and reasonable loading. They are more versatile than the simple AND/OR circuits and, therefore, are more widely used.
- 25. (b) binary 1. The output of a NAND is binary 0 if, and only if, all inputs are binary 1. With a binary 0 on any or all inputs, the output is binary 1.
- 26. (a) binary 0. The output of a NOR is binary 0 if any one or more inputs is binary 1.
- 27. (c) $C = \overline{A \cdot B}$ is the output expression for a NAND.
- 28. (d) $F = \overline{D + E}$ is the output expression for a NOR.

How NAND/NOR Gates Are Used.

The AND and OR logic functions can be performed by connecting inverters on the outputs of NAND and NOR gates respectively. Since the NAND and NOR functions are simply complementary logic functions of basic AND and OR operations, then it is logical to assume that the AND and OR operations can be obtained from NAND and NOR gates simply by adding an additional inversion. As stated in the previous section on inverters, the effect of one inversion is cancelled by adding a second. Cascading an even number of inverters removes the inversion function.

Figure 3-34 shows how the AND and OR operations are performed with NAND and NOR gates. But now, how is the OR function performed with a NAND gate and the AND function performed with a NOR gate?

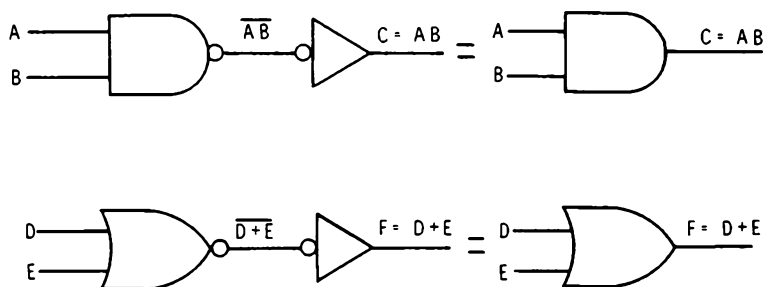


Figure 3-34

Performing AND and OR
operations with NAND and NOR gates.

As mentioned earlier, any of the three basic logic functions – AND, OR, NOT – can be performed by either a NAND gate or a NOR gate. You have already seen how the AND and OR functions can be obtained with NAND and NOR gates by simply inverting the output. To obtain the other logic functions with each type of gate, inverters are used on the inputs.

INVERTED INPUTS		NAND INPUTS		OUTPUT
X	Y	A	B	C
1	1	0	0	1
1	0	0	1	1
0	1	1	0	1
0	0	1	1	0

Figure 3-35

Truth Table for NAND gate.

Figure 3-35 shows the truth table for a NAND gate. The inputs are A and B and the output is C. Now consider the effect of adding an inverter to each input as shown in Figure 3-36. These inverters simply complement the input signals. The effect is illustrated in the truth table of Figure 3-35. The input signals to the inverters are labeled X and Y. Note that they are the complements of the signals A and B respectively. Considering the two inverters and the NAND gate as forming a single composite circuit, our inputs become X and Y instead of A and B. Output C remains the same. By observing the truth table in Figure 3-35 with this change in mind, you can now evaluate the logic function of the circuit. Disregarding the sequence of the X and Y input combinations and the A and B inputs, note the output state for each of the input states. You can see that the circuit produces a binary 1 output any time a binary 1 is applied to either one or both of the inputs. By definition this is the logic OR function. The OR function can be performed with a NAND gate by simply placing inverters ahead of each input.

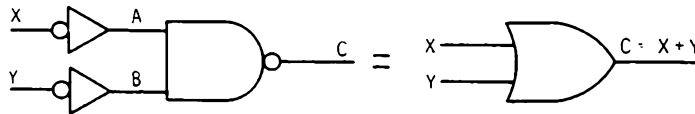


Figure 3-36

Performing the
OR function with a NAND gate.

By using inverters at the input of a NOR, the AND function can be performed. Figure 3-37 shows the truth table for a NOR gate. The inputs are D and E and the output is F. Now assume that inverters are connected ahead of the NOR gate and the new inputs are labeled L and M as shown in Figure 3-38. The inverter or complementary input states would be as indicated in the truth table of Figure 3-37. If we interpret the logic function of the composite circuit where the inputs are L and M and the output is F, then you can see by studying the truth table that the AND function is being performed. Note that the only time that the output F is a binary 1 is when both inputs L and M are at the binary 1 level. By definition this is the AND logic function. As Figure 3-38 indicates, a NOR gate with inverters at its input performs the AND function.

As you can see any of the three basic logic functions can be performed with either a NAND gate or a NOR gate. For that reason entire digital circuits can be constructed with just one type of gate. The choice is arbitrary and strictly up to the designer. There are some cases where both NANDs and NORs are mixed within a circuit. As you will see later, some circuit economies result when NAND and NORs are combined. However, just remember that any logic function can be implemented with NAND gates or NOR gates alone.

INVERTED INPUTS		NOR INPUTS		OUTPUT
L	M	D	E	F
1	1	0	0	1
1	0	0	1	0
0	1	1	0	0
0	0	1	1	0

Figure 3-37

Truth Table for NOR gate.

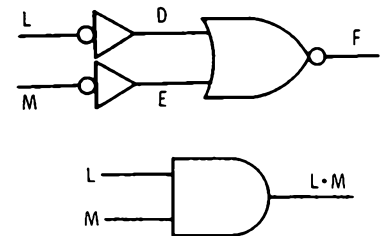


Figure 3-38

Using a NOR gate
to perform the AND function.

Self Test Review

29. A NAND or NOR gate can perform any of the three basic logic functions AND, OR, and NOT.
- a. True
 - b. False
30. When inverters are used at the inputs to a NOR gate, the resulting circuit performs what logic function?
- a. AND
 - b. OR
 - c. NAND
 - d. NOR
 - e. NOT
31. When inverters are used at the input to a NAND gate, the circuit performs what logic function?
- a. AND
 - b. OR
 - c. NAND
 - d. NOR
 - e. NOT
32. A NOR or NAND can be used as a NOT circuit by
- a. inverting the output
 - b. inverting the inputs
 - c. connecting all inputs together
 - d. cascading an even number of circuits.
33. Which circuit is preferred in implementing an entire circuit with one type of gate?
- a. NAND
 - b. NOR
 - c. Either

Answers

- 29. (a) True
- 30. (a) AND
- 31. (b) OR
- 32. (c) connecting all inputs together. By applying the same signal to all inputs of either a NAND or a NOR circuit it will perform the NOT or complement function.
- 33. (c) Either. NANDs or NORs can be used to implement a complete logic circuit. One type works as well as the other.

PRACTICAL LOGIC CIRCUITS

Now that you have completed this overview of the types of logic circuits in common use in digital equipment, let's take a look at several typical ways these logic elements are implemented. There are many different ways of electrically or mechanically obtaining the particular characteristics specified by the various types of logic elements. Here we will consider several popular ways of realizing digital logic elements with hardware.

Relays and Switches

The three basic logic functions – AND, OR, and NOT – can be readily implemented with relay contacts and switches. For example, Figure 3-39 shows an AND gate made with relays. Here the normally open (N.O.) contacts of two relays labeled A and B are connected in series with a battery and a lamp. In this circuit, a closed relay contact and an ON lamp represent a binary 1. An open contact and an OFF lamp represent a binary 0. A zero voltage level represents binary 0 and a positive voltage level represents a binary 1 on the relay coil. You can see, for any of the four possible input combinations there is only one where the lamp will light. If either one or both of the relays are de-energized, their contacts will be open and current will not be supplied to the lamp. However, if voltage is applied to both relay coils, both contacts will be closed and a binary 1 (ON lamp) output will occur. Contacts A and B must be closed to light the lamp. Series connected switches usually perform the AND function.

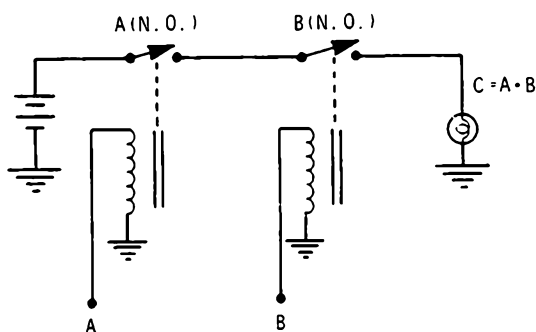


Figure 3-39

Relay AND gate.

Figure 3-40 shows an OR gate made with relays. The normally open relay contacts are connected in parallel. Here you can see that when a binary 1 voltage level is applied to either or both relay coils the D or E contacts will close, thereby supplying voltage to the lamp. An input to either relay D or E will close the circuit and turn on the lamp, representing a binary 1. Parallel connected contacts usually perform the OR operation.

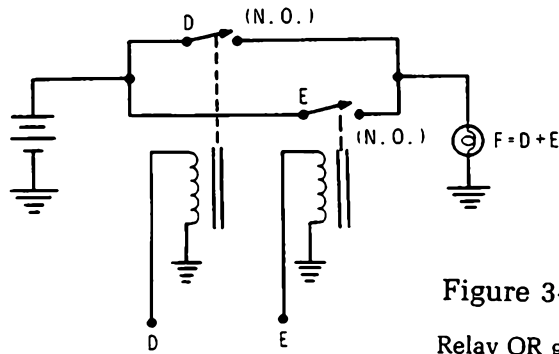


Figure 3-40

Relay OR gate.

A relay logic inverter circuit is shown in Figure 3-41. Here a relay with normally closed (N. C.) contacts is used. With a binary 0 (zero volts) applied to the relay coil, the relay is de-energized and the contact is closed representing a binary 1. This connects the battery to the lamp causing it to light. Therefore, with a binary 0 input the output is binary 1. Applying a binary 1 voltage level to the relay coil will energize the relay and open the contacts. This will cause the lamp to go off indicating a binary 0.

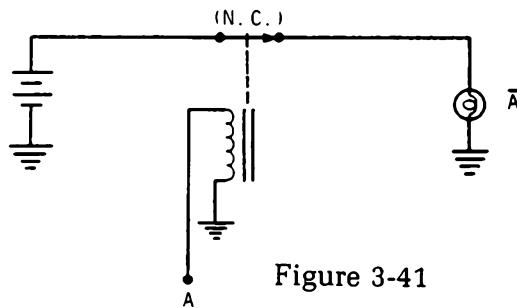


Figure 3-41

Relay inverter.

These relay switching circuits can be combined in many ways to form any logic function. In addition, manually operated switches can also be substituted for the relay contacts in some applications. Such relay or switch logic circuits are not often used today. Such circuits are large, slow in operation, and consume a significant amount of power. For most applications they are not practical. The very earliest of digital equipment, including some computers, were implemented with relays. However, many other different types of logic circuits are available now and these have many significant advantages over relays. There are still a few practical uses for relay and switch logic circuits. In some heavy industrial control systems where speed and power consumption is of little importance, relay logic circuits can handle high power applications and are very reliable. There are some applications where a mechanical means of operating the switches is available, thereby making mechanical or manual switching logic necessary or desirable.

Discrete Component Logic Circuits

A discrete component logic circuit is a logic element made up of individual electronic components such as transistors, diodes, resistors, capacitors, and other devices. These are assembled to form a complete circuit like the diode gates and inverter described earlier. For many years digital logic circuits were implemented with discrete components. They offered small sized, high performance, and reasonable power consumption. However, today such discrete component circuits are rarely used. Like relay and mechanical switching logic circuits they have essentially been replaced by logic elements with greater performance, lower cost, and improved features. You may still encounter discrete component logic circuits in some high power applications or in older digital equipment. Today, however, most digital logic functions are implemented with integrated circuits.

Integrated Circuits

An integrated circuit (IC) is a semiconductor device which combines transistors, diodes, resistors, and capacitors in ultra-miniature form on a single silicon chip. The advances in semiconductor technology have permitted the semiconductor manufactures to design, develop, and produce entire electronic circuits on a single silicon wafer that is generally less than one tenth of an inch square. These circuits are not only significantly smaller in size than discrete component logic circuits, but also offer many other benefits as well. Because they are mass produced, their cost is substantially less than discrete component circuits. Many offer significant savings in power consumption. Perhaps even more important is the elimination of the need for circuit wiring. When discrete components are used the components must be interconnected physically on a printed circuit board and then tested. With an integrated circuit the entire circuitry, all components included, are manufactured simultaneously. Manufacturing costs are reduced and reliability is improved.

Integrated circuits have been in existence since the late 1950's. During this time significant advances have been made. The complexity and sophistication of the circuits have increased significantly while the prices have continued to decline. Today, with integrated circuit techniques, it is not only possible to implement the basic logic elements, but also it is possible to fully integrate complete combinational and sequential circuits. Integrated circuits implementing the basic logic functions such as NAND, NOR, and flip-flops are known as small scale integrated circuits (SSI). Complete functional circuits of either the combinational or sequential type such as counters and decoders are generally designated as medium-scale integrated circuits (MSI). Complete circuits and systems, an entire computer memory for example, can also be constructed on a single chip and are known as large-scale integrated circuits (LSI), and even higher density circuits are also available. An entire microcomputer with CPU, memory, I/O, and related circuits can be obtained as a single integrated circuit. A complex device such as this is usually referred to as very large scale integrated circuit (VLSI).

Today most digital equipment is implemented with integrated circuits. This course emphasizes digital integrated circuits, their operation and application.

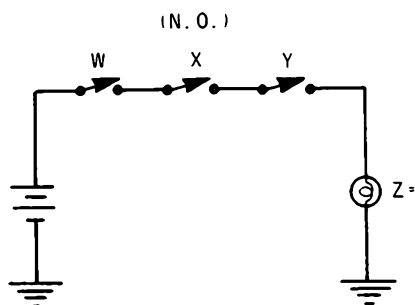


Figure 3-42

Circuit for Self-Test
Review Question 34.

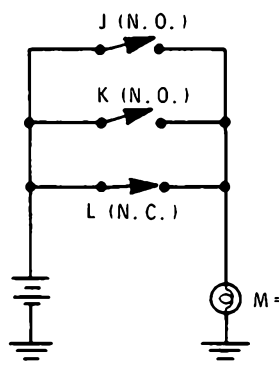


Figure 3-43

Circuit for Self-Test
Review Question 35.

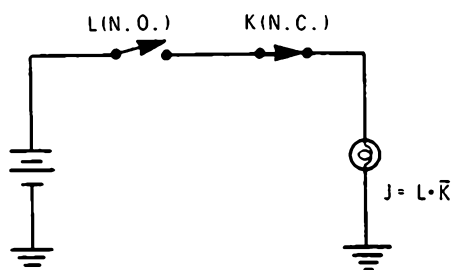


Figure 3-44

Answer to Question 36.

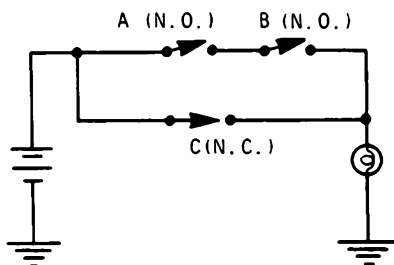


Figure 3-45

Answer to Self-Test
Review Question.

Self Test Review

34. Write the logic equation of the relay logic circuit shown in Figure 3-42.
35. Write the logic equation of the relay logic circuit shown in Figure 3-43.
36. Draw a relay logic diagram for the function $J = L\bar{K}$.
37. Draw a relay logic circuit for the function $D = A \cdot B + \bar{C}$
38. Some relay and discrete component logic circuits are still used in some applications requiring
 - a. high speed
 - b. low cost
 - c. small size
 - d. high power
39. The basic logic functions NAND, NOR, etc., in IC form are known as:
 - a. SSI
 - b. MSI
 - c. LSI
 - d. discrete

Answers

34. $Z = W \cdot X \cdot Y$
35. $M = J + K + \bar{L}$
36. See Figure 3-44.
37. See Figure 3-45.
38. (d) high power
39. (a) SSI small scale integration

EXPERIMENT 2

Logic Inverter

OBJECTIVE: *To demonstrate the operation and characteristics of typical discrete component and integrated circuit logic inverters.*

Materials Needed

- 1 – MPSA20 transistor (417-801)
- 2 – $1\text{k}\Omega$ resistor
- 2 – $4.7\text{ k}\Omega$ resistor
- 1 – SN74LS04N TTL hex inverter integrated circuit (443-755)
- Heathkit ET-3200 Digital Design Experimenter
- DC Voltmeter

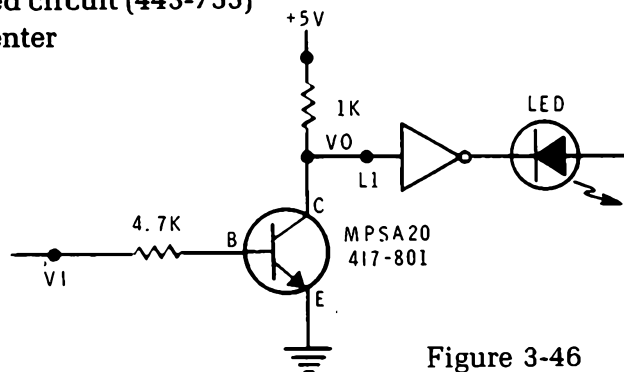


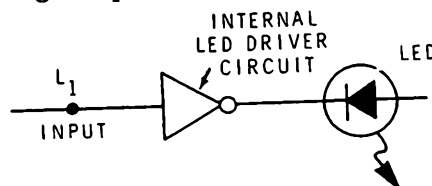
Figure 3-46

Procedure

1. Wire the circuit shown in Figure 3-46. The input is derived from logic switch SW1. You will measure the output state with your voltmeter and observe it on LED indicator L1.
2. Measure the inverter input and output voltages V_i and V_o with respect to ground for both positions of data switch SW1. Record your data in Table I.

NOTE:

The schematic diagram used to represent the logic indicators on the ET-3200 Digital Design Experimenter is shown below.



The inputs are labelled L1, L2, L3, and L4. The inverter symbol represents the internal LED driver circuit and not an external inverter. When the input is open or grounded, the LED is off. When the input is positive, the LED is on.

TABLE I

SW1 POSITION	V_i	V_o
DOWN		
UP		

3. From the data in Table I, answer the following questions.
- Does the circuit perform logic inversion? _____
 - What are the two output logic voltage levels assuming positive logic?
 binary 0 _____
 binary 1 _____
 - Make a truth table from the data in Table I. Use positive logic and assume the circuit input is A.
4. Disconnect the 4.7 k Ω base resistor from input SW1 and let it hang free. Measure the output voltage V_o . $V_o =$ _____. With an open input, the output is binary _____. An open input has the same effect as a binary _____ input. (Use positive logic.)
5. Connect the free end of the 4.7 k Ω base resistor to the CLK output and set the clock frequency to 1Hz. Connect LED indicators to the circuit inputs and outputs as shown in Figure 3-47. Observe the operation of the circuit by watching the LED indicator states and their relationship to one another. The circuit output is always the _____ of the input.

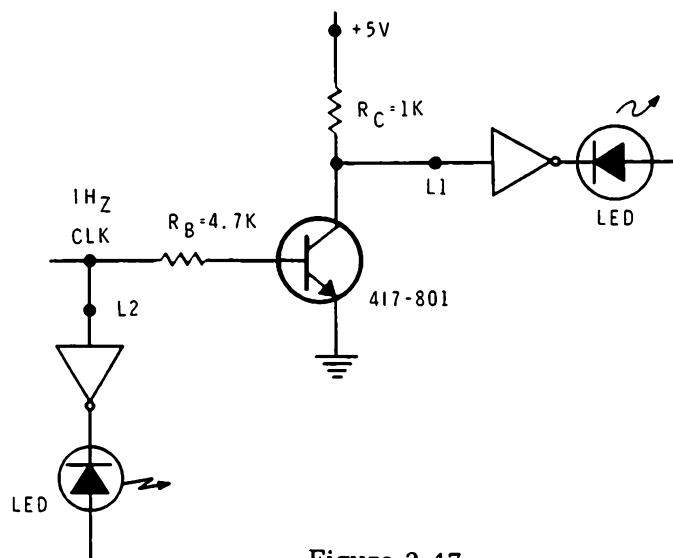


Figure 3-47

6. Modify your circuit as shown in Figure 3-48. Here you are cascading two inverter circuits. You will monitor the input on LED indicator L2 and the output of the second inverter on LED L1. Observe the input and output states as the 1 Hz clock operates the circuit.

If the input to the circuit in Figure 3-48 is binary 1, the output will be binary _____. If the output is binary 0, the input must be binary _____.

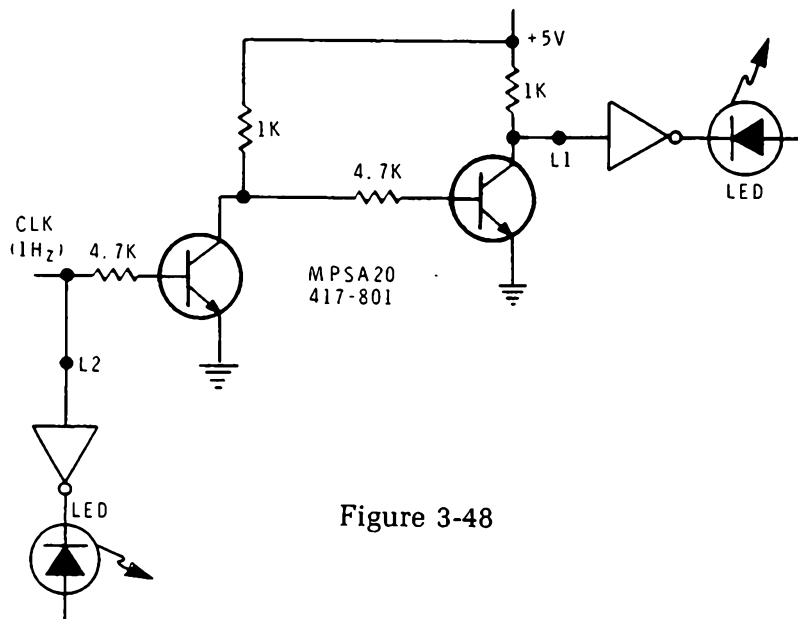


Figure 3-48

Discussion of Steps 1 Through 6

In Steps 2 and 3 you demonstrated inverter action. With input V_o equal to zero volts, (SW1 down), the emitter-base function is not forward biased so the transistor does not conduct. The output voltage V_o is +5 volts as seen through the 1K collector resistor. With +5 volts input (SW1 up), the transistor is saturated and the output voltage is about .1 volts. Inversion is performed. The positive logic levels are binary 0 = .1V and binary 1 = +5 volts. The logic truth table you constructed for this circuit should appear as shown in Table II below.

TABLE II

INPUT	OUTPUT
A	\bar{A}
0	1
1	0

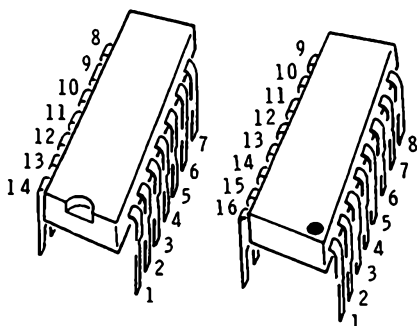


Figure 3-49

Typical 14- and 16-pin DIP ICs showing pin numbering schemes.

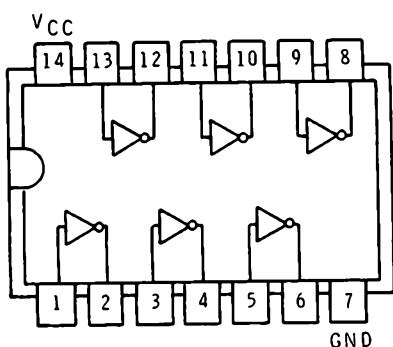


Figure 3-50

Top view of 74LS04 hex inverter IC showing pin assignments.

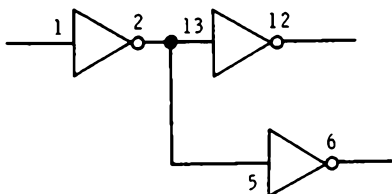


Figure 3-51

The input and output are complementary.

In Step 4 you disconnected the inverter input and left it open. With this condition, no forward bias is applied to the transistor so it does not conduct. The output voltage at the collector is +5 volts or binary 1. Therefore, an open input produces the same effect as a binary 0 input.

In Step 5 you further demonstrated logic inverter action. The 1 Hz clock was used to drive the inverter. You observed the input and output states on the LED indicators. You should have found that the input and output states were always the opposite of one another or complementary as indicated by L1 and L2 which alternately switch on and off at a 1 Hz rate.

In Step 6 you cascaded two inverter circuits and observed their operation. You should have found that the output of the second inverter is the same as the input. One inverter cancels the effect of the other. The output state is the same as the input state when an even number of inverters is cascaded.

Integrated Circuit Hex Inverter

In the steps to follow, you will be using a typical TTL (transistor transistor logic) integrated circuit (IC) to demonstrate inverter action. This IC and all others supplied in this program are housed in a plastic dual in-line package (DIP) as shown in Figure 3-49. Both 14- and 16-pin versions are available. Note the way in which the pins are numbered. A notch or dot (indentations, etc.) is used to designate the location of pin 1.

Figure 3-50 shows the logic diagram and physical pin connections (pin out) on the SN7404N (or simply 7404) hex inverter IC. This IC contains six identical and independent logic inverters. The inputs and outputs are identified by pin number. In the diagrams used in these experiments the package outline will not be shown. Instead, only the inverter symbol with pin number labeling will be given. See the example in Figure 3-51. In an experiment, if no pin numbers are given you can use any inverter you want. Not all of the inverters in the IC will be used in every experiment. Simply ignore those not used.

Refer to the instruction manual of your ET-3200 Digital Design Experimenter for details on installing the IC on the breadboarding socket. The IC should straddle the center notch and all pins should be seated firmly. The pins are delicate, so be careful when installing and removing the IC.

After the IC is installed, connect the power (V_{CC}) and ground leads. The supply voltage for the 7404 TTL IC is +5 volts and is applied at pin 14. Ground (GND) is connected to pin 7. Check the power and ground pin assignments for each IC you use as they vary from one type to another.

Procedure (continued)

- Mount a 74LS04 hex inverter IC (part number 443-755) on the breadboarding socket and connect pin 14 to +5 volts and pin 7 to ground (GND).
- Connect one of the inverters as shown in Figure 3-52. The input will come from logic switch SW1 and the output will be displayed on indicator L1.

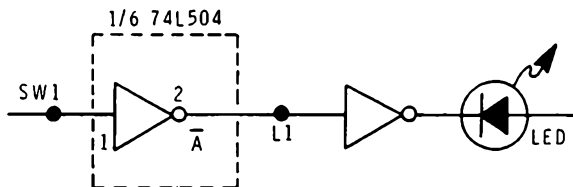


Figure 3-52

- Apply the logic voltages to the input as shown in Table III and measure the corresponding output voltages. Record in Table III.

Does the circuit invert? _____

What are the output logic levels for positive logic?

binary 0 _____ volts

binary 1 _____ volts

- Remove the connection between pin 1 of the IC and SW1 so that the inverter input is open. Measure the output voltage. With the input open, the output voltage is _____ volts, or for positive logic, a binary _____. This means that an open input has the same effect as a binary _____ input.

- Wire the circuit shown in Figure 3-53. How many inverters are cascaded? _____

TABLE III

A	\bar{A}
0V	
+5V	

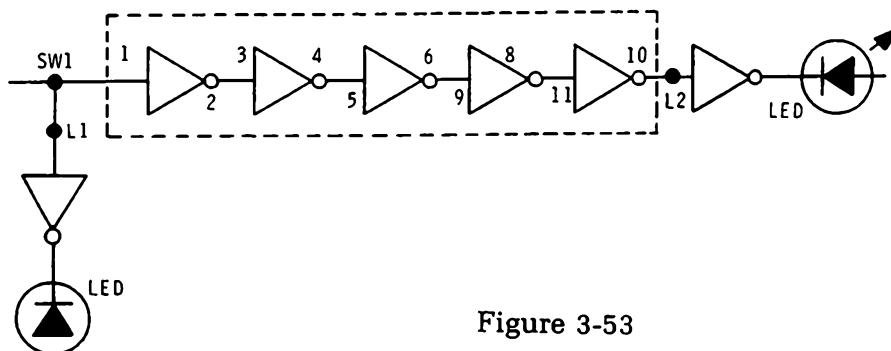


Figure 3-53

12. Set SW1 to binary 0 then binary 1 noting the output state for each input.
13. Modify your circuit to connect the L2 input of the LED indicator to pin 8 of the IC.

How many inverters are cascaded? _____

14. Apply binary 0 and binary 1 to the circuit with SW1 again noting the corresponding output state.

When an odd number of inverters are cascaded, the output is

- a. the same as
- b. opposite

the input. When an even number of inverters are cascade, the output is

- a. the same as
- b. opposite

the input.

Discussion of Steps 7 Through 14

In Steps 7, 8, and 9 you connected an integrated circuit inverter and evaluated its operation. With zero volts (binary 0) in you should have measured about +3.5 volts output (binary 1). With +5 volts input (binary 1) you should have measured about +.2 volts output (binary 0). Even with unequal input and output voltages, the circuit still performs logic inversion.

In Step 10 you determined the effect of an open input. With no input connection, the output is +.2 volts or binary 0. Since the input and output of a logic inverter are always complementary, the open input must be acting like a binary 1.

In Step 11 you cascaded 5 inverters. You should have found in Step 12 that the output was the complement of the input. Next, in Step 13, you monitored the output of the fourth inverter in the chain. In Step 14 you demonstrated that the input and output were the same. From this data, you can conclude that in a chain of inverters an odd number produces a complementary input and output, while with an even number of inverters, the input and output will be the same.

EXPERIMENT 3

Diode Logic Gates

OBJECTIVES: To demonstrate the operation and characteristics of diode AND and OR gates.

Materials Needed

- 2 – 1N4149 silicon diodes (#56-56)
- 1 – 1K ohm resistor
- 1 – 10 K ohm resistor
- 1 – DC Voltmeter
- 1 – ET-3200 Heathkit Digital Design Experimenter

Procedure

1. Wire the circuit shown in Figure 3-54. The inputs A and B come from data switches SW1 and SW2. You will measure the output voltage C with respect to ground.
2. Using data switches SW1 and SW2, apply the input voltages indicated in Table I to the logic gate. For each set of inputs, monitor the output voltage and record in Table 1.
3. Using positive logic assignments, convert the voltage levels in Table 1 into binary 1s and 0s and transfer to Table II.

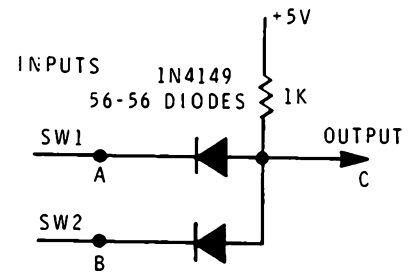


Figure 3-54

TABLE I

A	B	C
0V	0V	
0V	+5V	
+5V	0V	
+5V	+5V	

TABLE II

A	B	C

TABLE III

A	B	C

4. Study Table II and determine the logic function being performed.
Logic Function _____
5. Convert the voltage levels in Table I into 1s and 0s using negative logic assignments and transfer to Table III.

6. Study Table III and determine the logic function being performed.

Logic Function _____

7. Modify your experiment circuit so that it appears as shown in Figure 3-55. One logic input will come from SW1. The other logic input will be a 1 Hz clock (CLK) signal. You will monitor the gate output with LED indicator L1 and the clock output on L2.

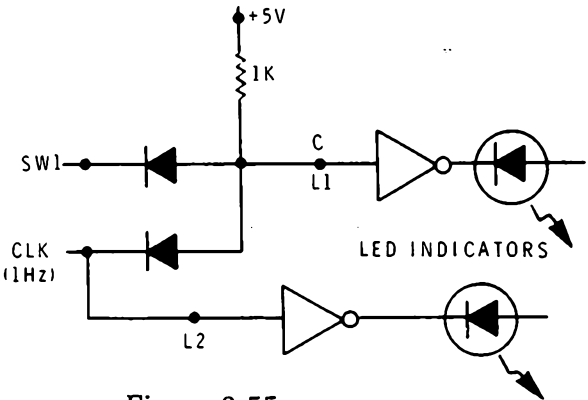


Figure 3-55

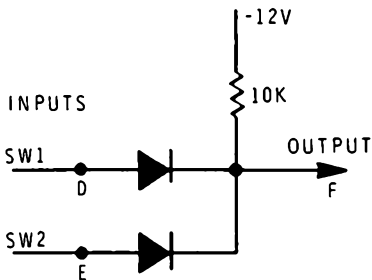


Figure 3-56

8. Set SW1 to Binary 0 (down). Note the output on LED L1. Then set SW1 to binary 1 (up). Again note the output on LED L1. Explain your results.

SW1 = 0 , Output C = _____

SW1 = 1 , Output C = _____

What logic function is the gate performing? _____

9. Construct the logic gate shown in Figure 3-56. Again, data switches SW1 and SW2 will supply the logic inputs D and E, and you will measure the output voltage at F with a DC voltmeter.

10. Apply the logic voltage levels indicated in Table IV to the circuit. Measure the output voltage for each set of inputs and record in Table IV.

TABLE IV

D	E	F
0V	0V	
0V	+5V	
+5V	0V	
+5V	+5V	

TABLE V

D	E	F

TABLE VI

D	E	F

11. Using positive logic assignments, convert the voltage levels in Table IV into binary 1s and 0s and transfer to Table V.

12. Study Table V and determine the logic function being performed.

Logic Function _____

13. Using negative logic assignments, convert the data in Table IV into binary 0s and 1s and transfer to Table VI.
14. Study Table VI and determine what logic function is being performed.

Logic Function _____

15. Modify your experimental circuit so that it appears as shown in Figure 3-57. One input is from logic switch A. The other input is a 1 Hz clock signal you will monitor with LED indicator L1. You will observe the clock output on LED indicator L2.

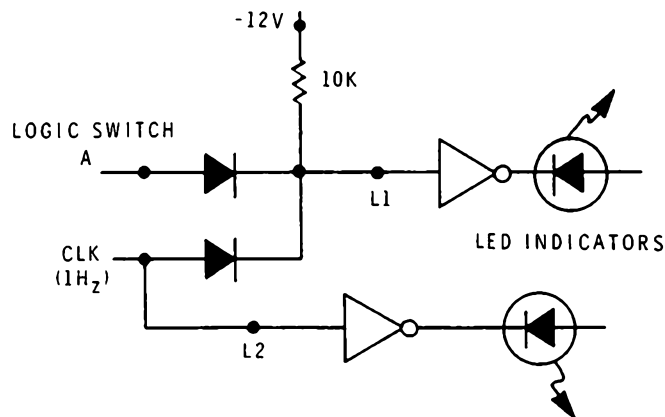


Figure 3-57

16. With logic switch A *not* depressed, the logic input to the gate is binary _____ (positive logic). Observe the gate output (L1) and clock (L2) signals. The gate output is _____.
 17. Depress logic switch A while observing the gate output. With A depressed the output is _____.
- What logic function is the gate performing? _____

Discussion

In this experiment you evaluated two basic types of diode logic gates. You determined their electrical performance by applying logic voltage inputs and measuring the corresponding outputs. Then, using both positive and negative logic level assignments, you determined the logic functions being performed. You also demonstrated several practical applications of these basic logic gates.

In Steps 1 through 6, you experimented with the gate in Figure 3-54. Your data in Table I, II, and III should appear as shown in Tables VII, VIII, and IX.

TABLE VII

A	B	C
0V	0V	+ .7V
+5V	0V	+ .7V
0V	+5V	+ .7V
+5V	+5V	+5V

TABLE VIII

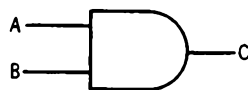
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

TABLE IX

A	B	C
1	1	1
1	0	1
0	1	1
0	0	0

From Table VII, you can see how the gate functions electrically. With either or both inputs at ground or zero volts, either one or both diodes conduct. The output, therefore, is the forward diode voltage drop of about .7 volts. When both inputs are +5 volts, neither diode conducts. The output is +5 volts as seen through the 1 k Ω resistor.

With positive logic assignments of the voltage levels in Table I (+5 volts = binary 1, 0V or +.7V = binary 0), the gate performs the AND function as you should have deduced from Table II. The output is a binary 1 only if all (both) inputs are binary 1. For all other input conditions, the output is binary 0. This is the AND function.



POSITIVE AND



NEGATIVE OR

Figure 3-58

Next, you evaluated the gate using negative logic assignments (+5 V = binary 0, 0V or +.7V = binary 1). From Table III, you should see that the OR function is being performed. The output is binary 1 (0V) if either one or both inputs are binary 1 (0V).

The logic gate in Figure 3-54 can perform both AND and OR operations. It is a positive AND/negative OR gate. The logic symbols representing these functions are indicated in Figure 3-58.

In Steps 7 and 8, you demonstrated a common application of a logic gate. Here, the gate acts as control elements to pass or inhibit the clock input. This control is handled by the SW1 input. With this input binary 0, the gate output is binary 0. The clock is inhibited and does not appear at the output. LED indicator L1 should have been off, indicating this condition. LED indicator L2 monitors the clock, so it should follow the 1 Hz pulsations.

With SW1 set to binary 1, the gate is enabled and the clock input is allowed to pass through to the output. LED indicators L1 and L2 should follow one another. Clearly, the AND logic function is being performed. This type of control gating is very widely used in digital circuits.

In Steps 9 through 14 you demonstrated the logic gate in Figure 3-40. You applied 0V and +5V input levels and measured the output for each combination to determine its electrical characteristics. Your results in Tables IV, V, and VI should be as shown in Tables X, XI, and XII below.

TABLE X

D	E	F
0V	0V	-.7V
0V	+5V	+4.3V
+5V	0V	+4.3V
+5V	+5V	+4.3V

TABLE XI

D	E	F
0	0	0
0	1	1
1	0	1
1	1	1

TABLE XII

D	E	F
1	1	1
1	0	0
0	1	0
0	0	0

With both inputs at zero volts, both diodes conduct and the output is the forward diode voltage drop of $-.7$ volts. If either one or both diodes conduct the output is the input logic level ($+5$ volts) less the forward drop of the conducting diode or approximately $+4.3$ volts.

Transferring this electrical data into binary 1s and 0s, you completed Tables V and VI. Studying Tables XI and XII, you can see that with positive logic assignments ($+4.3$ V or $+5$ V = binary 1, 0V or $-.7$ V = binary 0), the circuit performs the OR logic function since the output is binary 1 if either or both inputs is binary 1. With negative logic assignments ($+4.3$ V or $+5$ V = binary 0, 0V or $-.7$ V = binary 1), the gate performs the AND function. This gate is a positive OR/negative AND. The logic symbols representing these gate functions are shown in Figure 3-59.

Again, you demonstrated the dual nature of a logic gate. Any logic gate can perform either the AND or OR function depending upon the logic level assignments.

Finally, you demonstrated a useful application of an OR logic gate. In Figure 3-57, two logic inputs, A and the 1 Hz clock, drive the gate. With logic switch A *not* depressed, its output is binary 0. The gate output follows the clock input. With A depressed, its output is binary 1. This turns on the output indicator and keeps it on as long as A is held down. When the clock input is binary 1, the output is binary 1. When A is binary 1, the output is binary 1. The gate is performing the OR function.



POSITIVE OR



NEGATIVE AND

Figure 3-59

EXPERIMENT 4

Transistor Logic Gate

OBJECTIVE:

To demonstrate the operation and characteristics of a typical discrete component transistor logic gate.

Materials Needed

- 1 – MPSA20 transistor (417-801)
- 1 – 1K ohm resistor
- 2 – 4.7K ohm resistors
- DC Voltmeter
- Heathkit ET-3200 Digital Design Experimenter

Procedure

1. Construct the circuit shown in Figure 3-60. The inputs A and B will come from data switches SW1 and SW2. You will monitor the output on LED logic indicator L1.

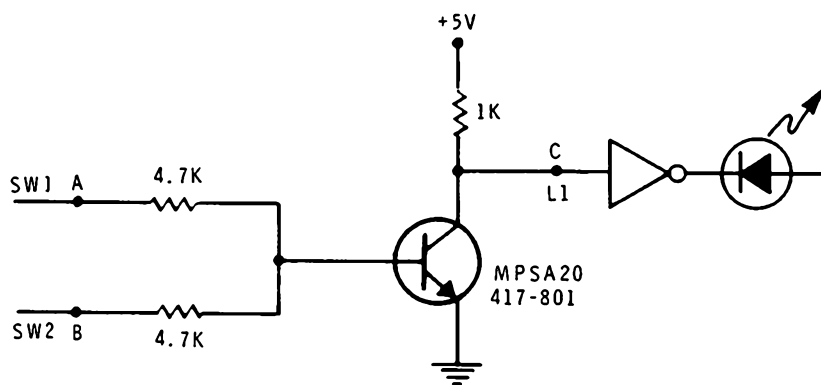


Figure 3-60

2. Apply the four input combinations given in Table I and measure the output voltage at C for each. Record your results in the C column in Table I.

TABLE I

A	B	C
0V	0V	
0V	+5V	
+5V	0V	
+5V	+5V	

3. Using positive logic level assignments, convert the data in Table 1 into a truth table using binary 0s and 1s. Use Table II.

TABLE II

A	B	C

4. Study Table II and determine the logic function being performed.
Logic function _____
5. Using negative logic level assignments, convert the voltages in Table I into binary 1s and 0s and complete Table III.

TABLE III

A	B	C

6. Study Table III and determine the logic function being performed.
Logic function _____

Discussion

The logic circuit in Figure 3-60 is called a resistor-transistor logic (RTL) gate. It is identical to a simple transistor inverter, but with two input base resistors. A positive voltage level applied to either or both of the inputs will saturate the transistor and cause the output to go low. With both inputs near zero volts or ground, the transistor will be cut off and the output will be the supply voltage as seen through the collector resistor. This operation is completely defined by your truth table (Table I) and should appear as shown in Table IV below.

TABLE IV

A	B	C
0V	0V	+5V
0V	+5V	+0.1V
+5V	0V	+0.1V
+5V	+5V	+0.1V

Using positive logic, your truth table (Table II) should be as indicated in Table V.

TABLE V

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

This defines the NOR function. The RTL gate acts as an OR gate followed by an inverter.

Using negative logic, your truth table (Table III) should appear as shown in Table VI.

TABLE VI

A	B	C
1	1	0
1	0	1
0	1	1
0	0	1

This is the NAND function. The gate performs the same function as an AND gate followed by an inverter for negative logic.

This circuit can be used to implement any of the basic logic functions by combining a number of gates. Additional base resistors can be added as more inputs are required. By paralleling inputs or using a single input, the circuit is nothing more than a simple inverter.

This type of logic gate was widely used in discrete component digital systems and a modified version of it is available in integrated circuit form.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this Unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and answer all of the questions first before checking your answers.

1. Which of the following is **not** one of the three basic logic elements?
 - a. AND
 - b. NAND
 - c. OR
 - d. NOT
2. A decision-making logic circuit with multiple outputs are a function of its multiple inputs and internal circuitry is called a:
 - a. logic element
 - b. logic gate
 - c. sequential circuit
 - d. combinational circuit
3. A logic circuit featuring flip-flop memory is usually designated a:
 - a. storage element
 - b. storage gate
 - c. sequential circuit
 - d. combinational circuit
4. Draw switch contact logic circuits representing the equations:
 - a. $M = L \cdot E \cdot F$
 - b. $X = P + R + S + T$
5. Three inverters are cascaded. The input is binary 1. The output is:
 - a. binary 0
 - b. binary 1
6. Analyze the operation of the circuit in Figure 3-61 using 0 (binary 0) and +5 volt (binary 1) logic levels. The circuit performs which logic function?
 - a. NAND
 - b. AND
 - c. OR
 - d. NOR

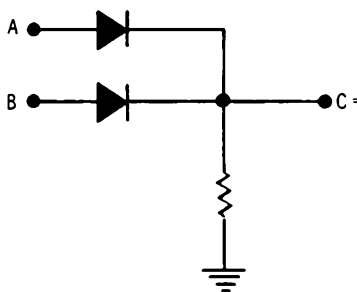


Figure 3-61

Circuit for exam Question 6.

7. Write the name of each logic symbol in Figure 3-62.

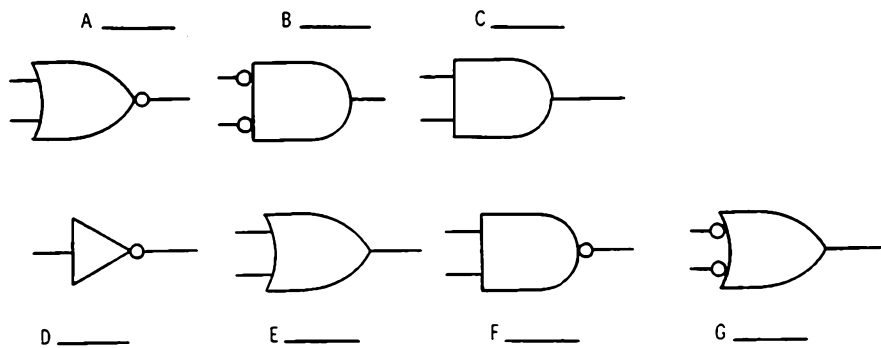


Figure 3-62

Illustration For Question 7

8. Fill in the logic function represented by the truth tables below.

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

A _____

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

B _____

D	E	F	G
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

C _____

9. A NAND gate has eight inputs. How many different input combinations are possible?

- a. 8
- b. 16
- c. 128
- d. 256

10. Name the logic function indicated by each equation below.

a. $K = B \cdot H \cdot D$ _____

b. $ORG = \overline{P} + F + N + S$ _____

c. $ADR = \overline{A0} \cdot A1 \cdot A2$ _____

d. $SNT = \overline{B}$ _____

e. $BS = \overline{JT + RW}$ _____

11. Refer to Figure 3-63. Draw the outputs you would expect from an AND gate and an OR gate with these inputs.

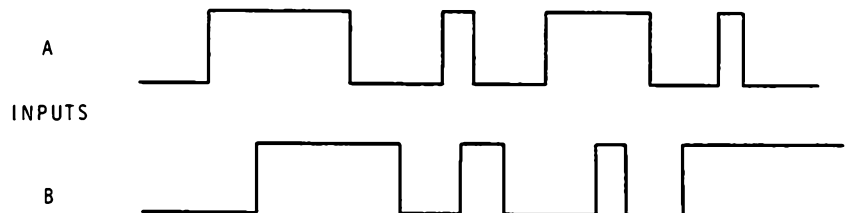


Figure 3-63

Illustration For Question 11

AND
OUTPUT _____

OR
OUTPUT _____

12. Analyze the operation of the circuit shown in Figure 3-64. Assume input levels of 0 (binary 0) and +3.6 volts (binary 1). This circuit has the output equation:

a. $A + B$

b. $A \cdot B$

c. $\overline{A \cdot B}$

d. $\overline{A + B}$

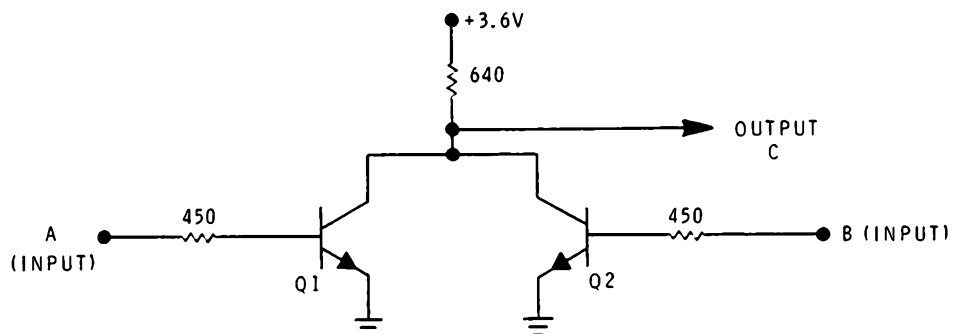


Figure 3-64

Figure For Question 12

13. A. The circuit in Figure 3-64 performs the positive logic function:
- a. OR
 - b. NOR
 - c. AND
 - d. NAND
- B. The circuit in Figure 3-64 performs the negative logic function:
- a. OR
 - b. NOR
 - c. AND
 - d. NAND
14. With negative logic assignments, (open = binary 1, closed = binary 0), series connected contacts perform the logic function:
- a. AND
 - b. OR
 - c. NAND
 - d. NOR

and parallel contacts perform the logic function:

- e. AND
 - f. OR
 - g. NAND
 - h. NOR
15. Draw the logic diagrams showing how to implement the AND and OR functions with a NAND gate.

EXAMINATION ANSWERS

1. b.—NAND The three basic logic elements are AND, OR and NOT.
2. d.—combinational circuit
3. c.—sequential circuit
4. See Figure 3-65

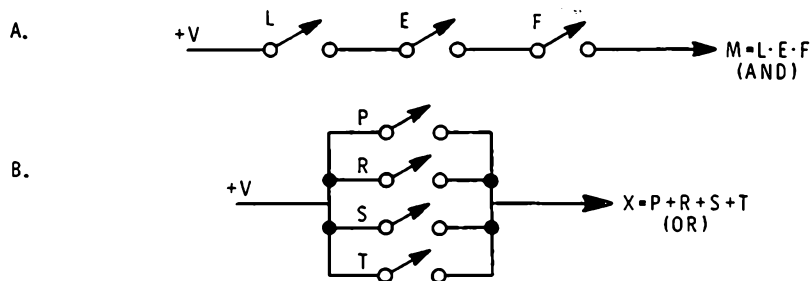


Figure 3-65

Solutions To Question 4

5. a.—binary 0
6. c.—OR The circuit in Figure 3-61 is a diode gate that performs the OR function with positive logic.
7. A.—NOR
B.—negative NAND
C.—AND
D.—inverter or NOT
E.—OR
F.—NAND
G.—Negative NOR
8. A.—NAND
B.—AND
C.—NOR
9. d.—256 $2^8 = 256$ With eight inputs there are 256 possible different combinations, 00000000 through 11111111.
10. a.—AND
b.—OR
c.—NAND
d.—invert
e.—NOR

11. See Figure 3-66

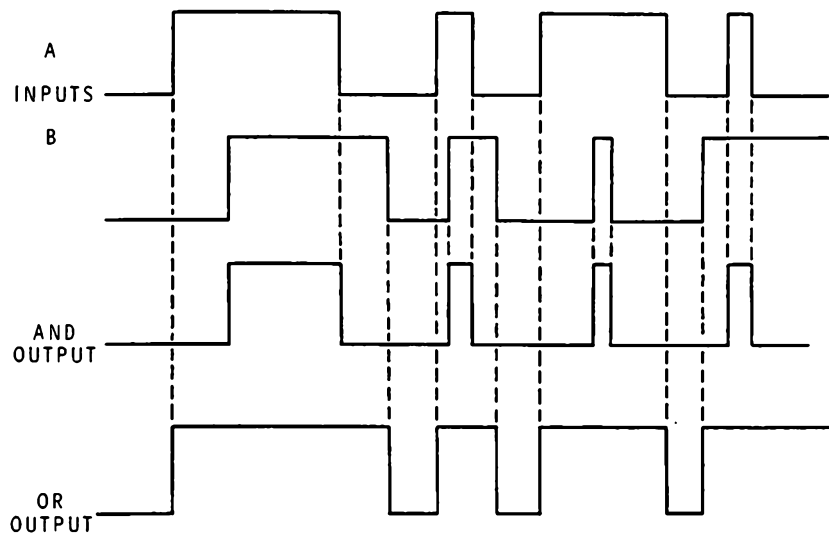


Figure 3-66

Solutions To Question 11

12. d.— $\overline{A+B}$
 13. A. b.—NOR
 B. d.—NAND

The circuit of Figure 3-64 is a resistor transistor logic (RTL) gate. If either or both inputs are +3.6 volts, the appropriate transistor conducts and the output is zero volts. If both inputs are zero, the output is +3.6 volts. The circuit performs the positive NOR function.

14. b.—OR
 e.—AND
 15. See Figure 3-67

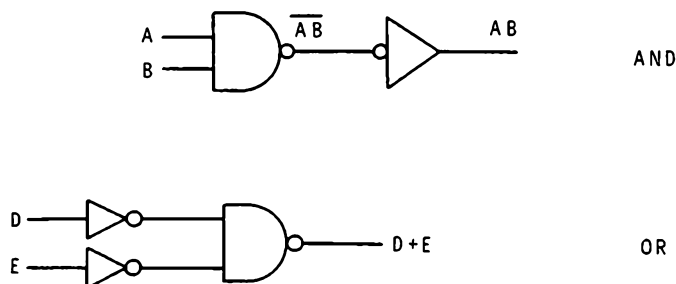


Figure 3-67

Solution To Question 15

APPENDIX

Positive and Negative Logic
Equivalent Circuits

POSITIVE LOGIC

INPUTS		OUTPUT
0V	0V	0V=0
0V	+5V	0V=0
+5V	0V	0V=0
+5V	+5V	+5V=1

TRUTH TABLE FOR
POSITIVE LOGIC
AND

INPUTS		OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

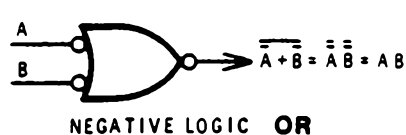
TRUTH TABLE FOR
AND GATE

NEGATIVE LOGIC

INPUTS		OUTPUT
+5V	+5V	+5V=0
+5V	0V	+5V=0
0V	+5V	+5V=0
0V	0V	0V=1

TRUTH TABLE FOR
NEGATIVE LOGIC
OR

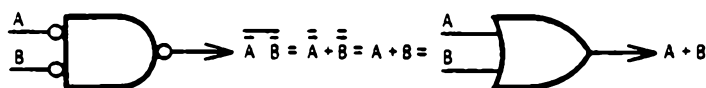
← EQUIVALENT CIRCUITS →



Both gates provide a 1 output only when all inputs are 1.

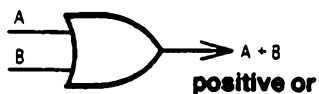
EXAMPLE 1.

To convert to negative logic, invert all inputs and the output.

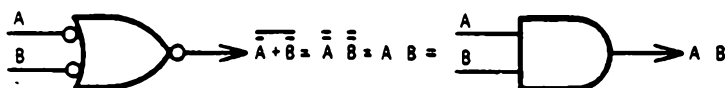


A negative logic AND is a positive logic OR.

EXAMPLE 2.



To convert to negative logic, invert all inputs and the output.

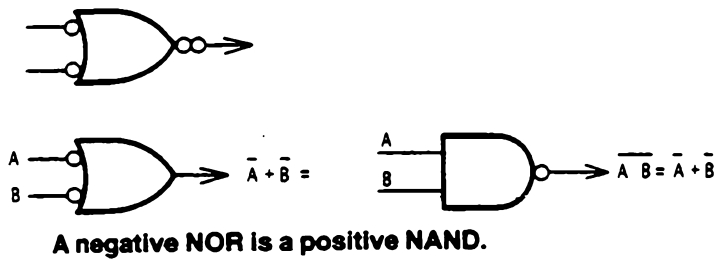


A negative logic OR is a positive logic AND.

EXAMPLE 3.

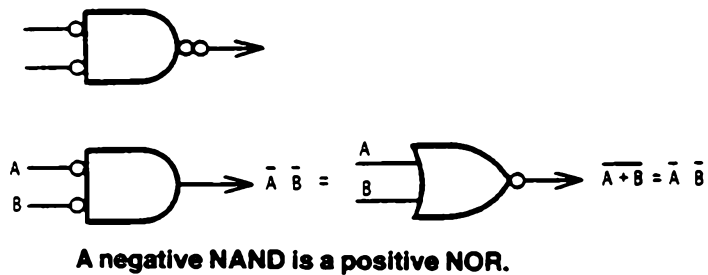
To convert to negative logic, invert all inputs and the output.

The two output inverters cancel each other. Thus, the symbol becomes:

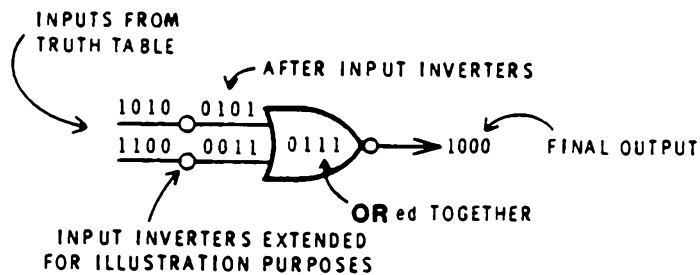
**EXAMPLE 4.**

To convert to negative logic, invert all inputs and the output.

Again, the two output inverters cancel and the symbol becomes:



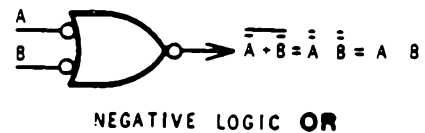
EXAMPLE 5.



INPUTS		OUTPUT
+5V=0	+5V=0	Hi=0
+5V=0	0V=1	Hi=0
0V=1	+5V=0	Hi=0
0V=1	0V=1	Lo=1

ACTIVE LOW

NEGATIVE LOGIC TRUTH TABLE



This circuit provides an output only when both inputs are active. From the truth table, you can see that the inputs are active low. Thus, the term negative logic means active low.

POSITIVE LOGIC

FUNCTION

SYMBOL

AND



OR



NAND



NOR



NOT



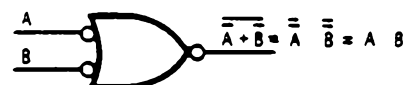
NEGATIVE LOGIC

FUNCTION

SYMBOL

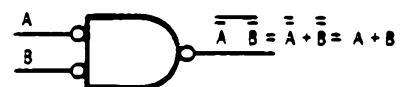
SAME AS

OR



SAME AS

AND



SAME AS

NOR



SAME AS

NAND



SAME AS

NOT



DE MORGAN EQUIVALENT CIRCUITS

Unit 4

**DIGITAL INTEGRATED
CIRCUITS**

CONTENTS

Introduction	4-3
Unit Objectives	4-4
Unit Activity Guide	4-5
Logic Circuit Characteristics	4-6
Integrated Circuits	4-17
Transistor Transistor Logic	4-25
Emitter Coupled Logic	4-38
Metal Oxide Semiconductor Integrated Circuits	4-42
Integrated Injection Logic	4-49
Selecting a Digital Integrated Circuit For A Specific Application	4-56
Experiment 5 — TTL Logic Gates	4-60
Experiment 6 — CMOS Logic Gates	4-66
Unit Examination	4-73
Examination Answers	4-77

INTRODUCTION

All modern digital equipment is constructed with integrated circuits. A study of digital techniques, therefore, is a study of digital integrated circuits and their application. Because of integrated circuits, digital equipment can be analyzed and designed almost entirely at a conceptual logical or systems level as opposed to an electronics or circuits level. Digital integrated circuits are, in the true sense of the word, building blocks which are used to construct digital equipment. Previously, the designer of digital equipment had to design not only the logic involved but also the electronic circuits necessary to implement that logic. With integrated circuits, the designer's job is primarily that of selecting commercially available devices and applying them to his specific application. No knowledge of electronic circuit design is necessary to the understanding and use of most digital techniques. However, by understanding the basic components and circuits used in modern digital integrated circuits your ability to work with and use them will be greater.

In this unit you will study the basic components and circuits used in the most common types of digital integrated circuits. The information in this unit will help you to analyze the operation of digital circuits and will help you to select a type of integrated circuit for a specific application. Look closely at the Unit Objectives that follow to determine the specific knowledge and skills you will have when you complete this unit. Then follow the steps in the Unit Activity Guide, recording the time you spent on each activity.

UNIT OBJECTIVES

When you complete this unit, you will have the knowledge and skills indicated below. You will be able to:

1. Name the two basic types of semiconductor switching elements used in digital circuits.
2. Define the four basic logic circuit characteristics of propagation delay, power dissipation, noise immunity and fan out.
3. Name and visually indentify the three basic types of digital IC packages.
4. Name four distinct families of digital ICs.
5. Explain the difference between current source and current sink types of logic circuits.
6. Describe the detailed operation and capabilities of TTL, ECL, MOS, CMOS, and IIL integrated circuits given a schematic diagram of the circuit.
7. Select a type of digital IC to implement a given application for optimum performance and economy.

UNIT ACTIVITY GUIDE

	Completion Time
<input type="checkbox"/> Read "Logic Circuit Characteristics."	_____
<input type="checkbox"/> Answer Self Test Review Questions 1–6	_____
<input type="checkbox"/> Read "Integrated Circuits."	_____
<input type="checkbox"/> Answer Self Test Review Questions 7–12.	_____
<input type="checkbox"/> Read "Transistor Transistor Logic."	_____
<input type="checkbox"/> Answer Self Test Review Questions 13–20.	_____
<input type="checkbox"/> Read "Emitter - Coupled Logic."	_____
<input type="checkbox"/> Answer Self Test Review Questions 21–25.	_____
<input type="checkbox"/> Read "Metal Oxide Semiconductor Integrated Circuits."	_____
<input type="checkbox"/> Answer Self Test Review Questions 26–31.	_____
<input type="checkbox"/> Read "Integrated Injection Logic."	_____
<input type="checkbox"/> Answer Self Test Review Questions 32–35.	_____
<input type="checkbox"/> Read "Selecting a Digital Integrated Circuit for a Specific Application."	_____
<input type="checkbox"/> Answer Self Test Review Questions 36–38	_____
<input type="checkbox"/> Perform Experiment 5.	_____
<input type="checkbox"/> Perform Experiment 6.	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Review the Examination Answers.	_____

LOGIC CIRCUIT CHARACTERISTICS

There are many different types of digital integrated circuits available to implement digital equipment. Both saturated and unsaturated bipolar transistors as well as MOSFETs are used to implement a variety of logic circuits. Each type or family of digital integrated circuits has its own special capabilities and limitations. Their characteristics vary widely and the optimum circuit to use in a given application depends upon specific needs and requirements.

Some of the most important characteristics of digital integrated circuits are logic levels, propagation delay, power dissipation, noise immunity, and fan out. By understanding the meanings of these characteristics you can quickly compare, contrast and evaluate different IC logic families.

Logic Levels

Logic levels are the voltage values assigned to the binary 1 and binary 0 states for a given type of digital integrated circuit. Nominal values for the two levels are generally given, but in practice, the actual voltage levels may vary somewhat because of internal component tolerances, power supply variations, temperature, and other factors. Generally, the manufacturer will list maximum and minimum acceptable voltage values for the binary 0 and binary 1 levels.

It is important to know the logic levels for a given type of integrated circuit so that when you are working with the equipment you can readily identify input and output logic states by measuring the logic levels with a voltmeter or an oscilloscope. A knowledge of the logic levels will permit you to analyze the operation of a circuit or determine whether it is functioning properly.

Propagation Delay

The propagation delay is a measure of the speed of operation of a logic circuit. Speed of operation is one of the most important characteristic of a digital circuit. For most digital applications high speed operation is beneficial.

Propagation delay is the amount of time that it takes the output of a digital circuit to respond to the input level change. It is the accumulation of all of the rise times, delay times, and storage times associated with any logic circuit. When the input voltage changes from the binary 0 to binary 1 or from the binary 1 to binary 0 levels, the output of the logic circuit will respond at some finite time later.

Figure 4-1 illustrates propagation delay. Shown here is the input to a digital circuit and the corresponding output. The circuit could be an inverter, a NAND gate, or a NOR gate. A binary 0 to binary 1 transition causes a binary 1 to binary 0 transition at the output. Note that the output transition occurs a specific time after the input transition. This is the propagation delay. The propagation delay (t_p) is generally measured between the 50 percent amplitude points on the corresponding leading and trailing edges of the input and output pulses. Note also that there are also two types of propagation delay, the propagation delay occurring when the output changes from high to low (t_{pHL}) and the propagation delay that occurs on the low to high output transition (t_{pLH}). Because of the characteristics of the logic circuit, the propagation delays for the two types of level changes are generally different. They are of the same order of magnitude and close in value but nevertheless unequal.

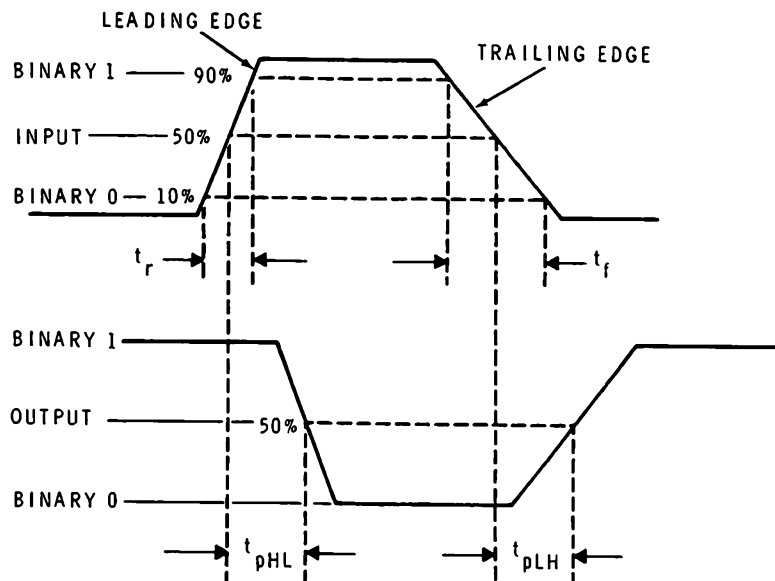


Figure 4-1
Propagation Delay.

The rise and fall times of the input and output pulses are another important consideration. The rise time (t_r) is the time it takes the pulse to rise from 10 percent to 90 percent of its maximum value. The fall time (t_f) is the time it takes for the pulse voltage to fall from 90 percent to 10 percent of its maximum value.

For most modern digital integrated circuits propagation delays are very short but finite. Propagation delays as low as 500 picoseconds are achievable. Some types of modern logic circuits have propagation delays as high as several hundred nanoseconds. Rise and fall times are usually less than the propagation delays. Because of manufacturing tolerances, circuit wiring and other factors, propagation delays can vary considerably from their nominal indicated value. In addition propagation delays are additive. When gates and other combinational logic circuits are cascaded, the propagation delays accumulate. If there is more than one level of logic, the total propagation delay from input to output is simply the sum of the individual gate propagation delays.

Power Dissipation

Another important characteristic of digital logic circuits is power dissipation. This is a measure of the amount of power consumed by the components in a typical logic gate or other circuit. Power dissipation, in milliwatts per logic gate, is an average value since the power consumption is usually different for the binary 1 and binary 0 output states.

The amount of power dissipated by a logic circuit is a very important consideration in the design of any digital equipment. A high power dissipation will mean high electrical energy consumption. Naturally, it is desirable to conserve as much electrical power as possible since the cost of operation of the equipment is an important consideration. This is particularly true of large scale digital systems such as computers.

The total power dissipation of the digital circuitry will also determine the size and cost of the power supply. In addition, high power dissipations mean high heat levels. In some instances special cooling requirements may be necessary to ensure proper operation of the equipment. Power dissipation is particularly important in portable or battery operated equipment. In order to reduce the cost of the battery and ensure long battery life, low power dissipation is desirable.

Gate power dissipation runs all the way from microwatts for certain types of MOS circuits to as high as 60 to 100 milliwatts per gate for certain types of high speed nonsaturated logic.

The Speed-Power Trade-Off

Two of the characteristics that we have considered so far, namely, speed and power dissipation are directly dependent upon one another in all types of digital logic circuits. The relationship between these two characteristics is such that speed is proportional to power dissipation. The faster a logic circuit switches the higher its power dissipation. In order to get high speed operation you must accept the penalty of high power dissipation. This trade-off or compromise between speed and power is one of the most important considerations that a digital designer must make in a selection of a type of logic circuit for a given application. High speed digital logic circuits use nonsaturating bipolar transistors. Because the transistors do not saturate, their emitter-collector voltage drops are higher. Combine this with the very low circuit resistance values to minimize charge and discharge times of stray capacitances and the result is high power consumption.

MOS integrated circuits consume a very small amount of power. Their high impedance nature is partially responsible for this, however, this characteristic plus the built-in capacitances make for very slow switching speeds. The result is that the frequency of operation is severely limited. Nevertheless, the extremely low power consumption – on the order of nanowatts – makes MOS circuitry extremely desirable for portable and battery operation where high speed is not required. Other types of logic circuits fall between these two extremes. The speed-power trade off is an inherent compromise.

You will sometimes see the speed-power relationship expressed with a number call the “speed-power product.” It is obtained by multiplying the propagation delay in nanoseconds by the power dissipation in milliwatts. The result is the speed-power product in picojoule. Joule is the unit of energy (power per unit of time) while pico means 10^{-12} . For example, a logic gate with a propagation delay of 5 nanoseconds and power dissipation of 3 milliwatts has a speed-power product of $5 \times 3 = 15$ picojoules. The lower the picojoule figure, the higher the quality of the circuit.

Noise Immunity

Noise immunity is a measure of the susceptibility of a logic circuit to noise pulses on the inputs and output of a logic circuit. Noise is considered to be any extraneous and undesired signal generated within the equipment itself or externally that is added to and appears superimposed upon the standard system logic levels. This noise can be a slowly varying dc level or very high frequency short duration voltage or current spikes. The noise may be either randomly occurring or repetitive. In any case the noise signals can cause the logic circuit to switch to an undesirable state at an improper time.

All digital logic circuits have built-in noise immunity. Because of the voltage thresholds associated with the components and the circuit, most logic circuits are capable of rejecting noise spikes of a relatively high amplitude. The noise immunity of most logic circuits is from approximately 10 to 50 percent of the supply voltage. This means that a noise spike occurring on a binary 0 or binary 1 level will be rejected if its amplitude is below a level that is 10 percent to 50 percent of the supply voltage. A circuit with a noise immunity of one volt, for example, would reject noise pulses that are one volt or less different from the nominal binary 0 or binary 1 logic levels. In some cases, noise is rejected by the logic circuit by virtue of its slow response. Some noise is high frequency in nature and noise pulses are of such short duration that the logic circuits cannot respond fast enough to cause a logic state change.

Noise immunity is an important consideration of digital logic circuits since most digital systems generate a substantial amount of noise during high speed switching. In addition, much digital equipment is used in noisy industrial environments where transients from the power line and other electrical equipment can cause false triggering of the logic circuitry. When selecting a particular digital integrated circuit for a logic application, noise immunity is an important consideration.

Fan Out

Fan out is a characteristic that indicates how much of a load can be connected to the output of a digital circuit. Fan out is generally expressed in terms of the number of standard size loads that a logic gate output can accommodate and still maintain proper operation at the nominal logic levels, speed, temperature range and other factors. Because of the component limitations and circuit configuration naturally there is a limit to the number of loads that can be connected to a logic circuit. A typical logic gate, for example, may have a fan out of ten, indicating that ten separate gate inputs can be attached to the output of this logic circuit and still maintain proper operation according to the manufacturer's specifications.

An important factor that has an affect on fan out of a logic gate is the current that flows into or out of the logic gate during the low and high status. In the following discussion, the conventional theory of current flow is used (current flow from plus to minus) because it demonstrates the principles of sinking and sourcing much clearer than electron flow.

Figure 4-2A shows the totem pole output stage of a TTL IC connected to a TTL input. When the output is logic low, Q_2 conducts and Q_1 is turned off. In this condition, the low impedance of Q_2 is between the load(s) and ground. This is the current sinking mode of operation. Using **conventional** current flow, current flows out of the emitter of the load to ground through sink transistor Q_2 . When the output is a logic high, Q_1 is conducting and Q_2 is turned off. This is the current sourcing mode of operation. Using **conventional** current flow, current flows from V_{CC} through Q_1 to the input emitter of the load. The magnitude of the current that flows during the logic high and low states is specified by the manufacturers and determines the fan out characteristics.

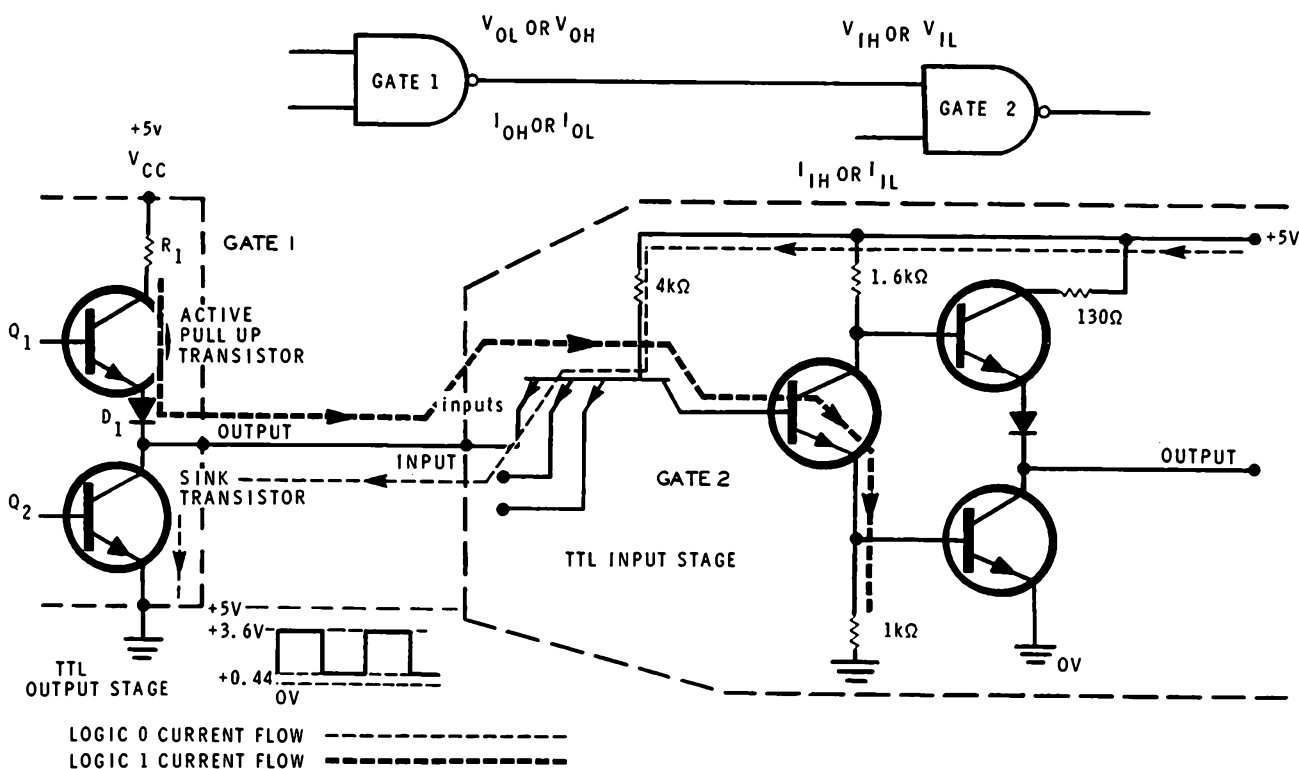


Figure 4-2A

Totem Pole TTL output driving a TTL input

For example, in the current-sinking mode, Q_2 of gate 1 must be able to sink current I_{IL} from each attached load. According to the TTL specification chart in Figure 4-2B, I_{TL} is 1.6ma for each standard TTL load. If the total sink current through Q_2 exceeds the 16ma (I_{OL}) shown in Figure 4-2B, the output voltage low will no longer be at 0.4 volts or less. The values shown in Figure 4-2B are for a standard TTL gate. The current values of I_{OL} and I_{OH} are worst case values which determine fan out.

LOGIC 1 STATE	LOGIC 0 STATE
I_{OH} - WILL SOURCE NO LESS THAN 400 μA	I_{OL} - WILL SINK NO MORE THAN 16 MA
V_{OH} - 2.4V OR HIGHER	V_{OL} - 0.4 VOLTS OR LESS.
I_{IH} - NO GREATER THAN 40 μA	I_{IL} - WILL SOURCE AT LEAST 1.6 MA
V_{IH} - 2VOLTS OR HIGHER	V_{IL} - NO GREATER THAN 0.8 VOLTS

Figure 4-2B

Standard TTL
worst case specifications

To simplify interconnection between different types of logic gates, manufacturers quote a normalized or standard load which is:

- $I_{IH} = 40\mu a$ maximum, 1 TTL unit load (U.L.) in logic 1 state.
- $I_{IL} = 1.6ma$ maximum, 1 TTL unit load in logic "0" state.

For example, if gate 2 of Figure 4-2A is a 7400 gate, which has, according to the manufacturer's data sheet, a maximum I_{IL} of 1.6ma and an I_{IH} of 40 μa , then gate 2 would have an input load factor of 1 U.L. (also known as a fan-in of 1). If gate 1 of Figure 4-2A is a 7400, it will sink 16ma in the low state and source 800 μa in the high state.

The output drive factors are:

$$\text{Output low } \frac{16 \text{ ma}}{1.6 \text{ ma}} = 10 \text{ U.L.}$$

$$\text{Output high } \frac{800 \mu\text{a}}{40 \mu\text{a}} = 20 \text{ U.L.}$$

This means that gate 1 of Figure 4-2A has a fan out of 10. This means it is capable of driving an additional 9 gates that have a fan in of 1, or any number of gates of other fan in values, but not exceeding the fan out value of 10.

Self Test Review

1. The most important logic circuit characteristics are:
 - a. _____
 - b. _____
 - c. _____
 - d. _____
 - e. _____
2. Typical propagation delay range for modern digital integrated circuits is
 - a. 1 to 100 milliseconds.
 - b. 1 to 100 microseconds.
 - c. 1 to 100 nanoseconds.
 - d. 1 to 100 picoseconds.
3. Decreasing the propagation delay of a logic circuit generally results in an increase in
 - a. power dissipation
 - b. fan out
 - c. noise immunity
 - d. package size
4. Increasing the number of loads on the output of a current source type logic circuit causes the binary 1 output level to
 - a. increase.
 - b. decrease.
 - c. remain the same.
5. Increasing the load on a current source type logic circuit causes more current to be drawn through the
 - a. collector resistor
 - b. output transistorand in current sinking logic increasing the number of loads causes more current to be drawn through the
 - c. collector resistor.
 - d. output transistor.
6. A logic circuit with a noise immunity of 40 per cent is better in rejecting noise than one with 10 per cent.
 - a. True
 - b. False

Answers

1. Logic levels, propagation delay, power dissipation, noise immunity and fan out.
2. (c) 1 to 100 nanosecond (1 nanosecond = 10^{-9} second)
3. (a) power dissipation
4. (b) decrease
5. (a) collector resistor
(d) output transistor
6. (a) True

INTEGRATED CIRCUITS

Since all modern digital equipment is made up of integrated circuits, you should be familiar with the various types. In this section you are going to learn how integrated circuits are classified and something about their physical characteristics. You will also learn of the most popular families of integrated circuits used in digital equipment today.

Integrated circuits are classified in three basic ways: by method of manufacturing, by application and by function. Let's briefly consider each of these type of classifications.

Manufacturing Methods

There are four basic ways of making integrated circuits. The most widely used method is called monolithic. Other types of manufacturing methods include thin film, thick film, and hybrid.

Monolithic. A monolithic integrated circuit is one that is constructed entirely on a single chip of silicon semiconductor material. Semiconductor materials are diffused into the basic substrate or base material to form the various junctions making up components such as diodes, transistors and resistors. The semi-conductor materials to be diffused into the substrate are in gaseous form and are deposited on the substrate through a series of masking operations under very high temperature. The result is that the entire circuit, all components and interconnections, are on a single base, thus the term monolithic. Most integrated circuits are constructed using this monolithic technique.

There are two basic forms of monolithic integrated circuits: bipolar and MOS. Here the difference is primarily that of the type of transistors used in constructing the circuits. Bipolar circuits which can be either saturating or non-saturating are by far the most widely used. But MOSFET circuitry is becoming more popular. The MOS circuitry is easier to make and takes up less space; therefore, much more circuitry can be placed on a silicon chip of a given size. The simplicity of the components also makes the manufacturing yield much higher. The result is that MOS circuits can be constructed with higher density and at lower costs.

Thin and Thick Film Techniques. Thin and thick film integrated circuits are manufactured by depositing certain materials on a non-conducting base such as ceramic. Through a series of masking procedures, various resistive and conducting materials are deposited on the base or substrate to form resistors, capacitors, and inductors. Semiconductors are not usually manufactured in this way. Thin and thick film techniques are primarily used for manufacturing passive networks such as attenuators, filters, phase shift networks and the like. Because such networks can be made extremely small, they offer the same advantages over discrete component circuits as do monolithic integrated circuits. Another advantage is that component tolerances can be closer than equivalent components made by monolithic techniques. For high quality precision circuits, thin and thick film techniques are preferred.

Hybrid Circuits. A hybrid integrated circuit is made up of a combination of monolithic, thin film or thick film circuits. Any number of combinations are considered to be hybrid. A hybrid integrated circuit may consist of multiple monolithic chips interconnected in a single package. Another example of a hybrid is a monolithic circuit combined with a thin film or thick film passive network. Sometimes monolithic circuits and thin film or thick film circuits are also combined with individual semiconductor component chips to form a special high grade circuit for an unusual application.

Hybrids offer the advantage that a variety of different integrated circuits and components can be combined to offer special advantages not available in individual types of integrated circuits alone. For example, because of the ultra small size of a monolithic circuit, power dissipation is limited. In order to handle high power requirements, it may be necessary to combine a low power monolithic circuit with a power transistor mounted on a separate chip, but physically interconnected within the same package. High precision circuit requirements might be met by a combination of a monolithic circuit and a highly accurate thin film network. Because more than one type of technique is involved, and the complication of the interconnections that are necessary, hybrid circuits are more complex and expensive than other types. However, they do offer the designer a wide range of capabilities while still maintaining the ultra small size and other benefits generally associated with integrated circuits. Hybrids are often used where it is necessary to combine analog (linear) and digital circuits.

Application

Another method of classifying integrated circuits is by their application. Primarily, this is a means of distinguishing between linear and digital circuits. Digital integrated circuits, of course, work with logic levels, pulses and binary data. Such switching circuits use either bipolar transistors or MOSFETs. Linear integrated circuits usually involve amplifiers of some kind and work with analog signals. They are constructed with bipolar transistors.

The chart in Figure 4-3 shows the basic integrated circuit hierarchy.

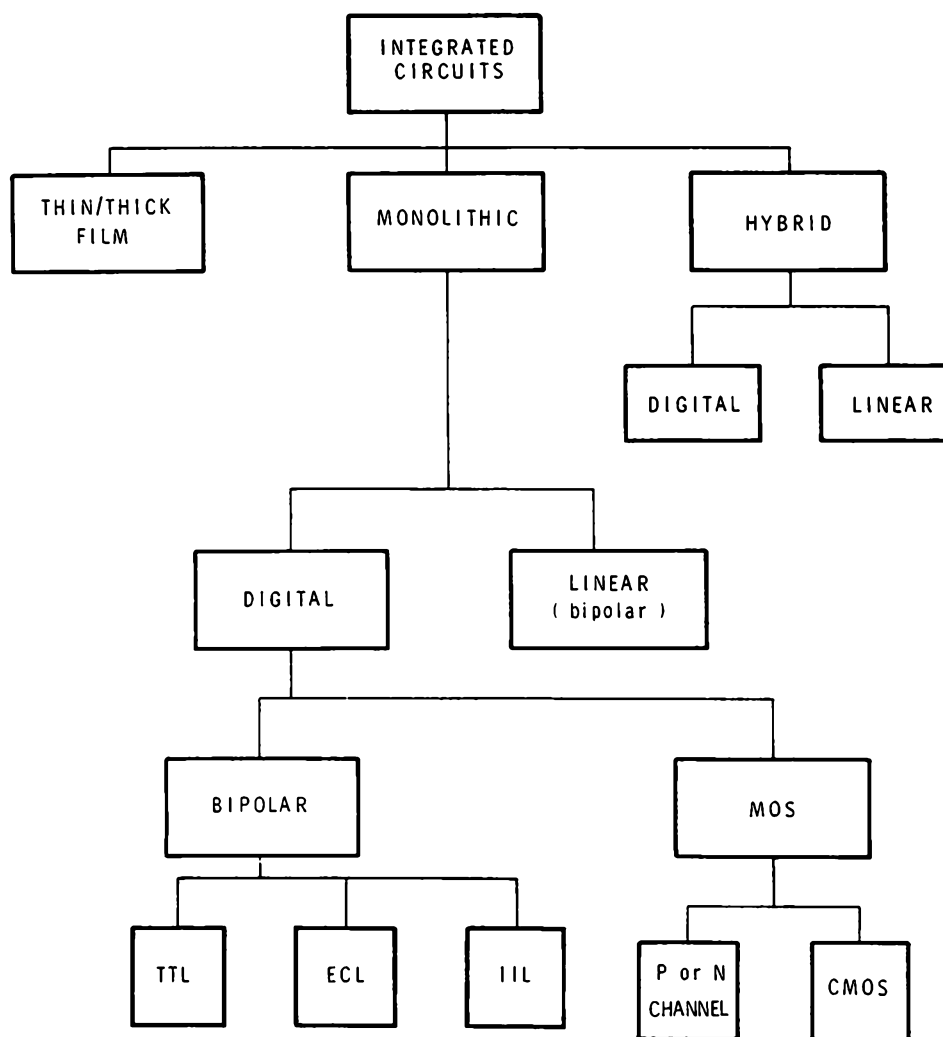


Figure 4-3

Hierarchy of integrated circuits.

Function

There are four basic classifications that identify the function of a digital integrated circuit: small scale integration (SSI), medium scale integration (MSI), large scale integration (LSI), and very large scale integration (VLSI).

SSI circuits are the simplest and most basic form of integrated circuits. These are amplifier or gate circuits that perform a single basic function. They must be interconnected externally in order to form complete functional or operational circuits. A typical SSI digital integrated circuit might consist of several multiple input gates or a flip-flop.

Medium scale integrated circuits are more complex. MSI circuits involve multiple gates which are interconnected to form a complete functional circuit. Most MSI circuits contain twelve or more equivalent gates or circuitry or similar complexity. An MSI circuit is usually a complete functional operating network such as a decoder, a counter or multiplexer. Such circuitry eliminates the need of having to interconnect individual gates in SSI packages to form the same function. MSI circuits greatly reduce the number of integrated circuits in a system and thereby reduce cost, assembly time, and in some cases, power consumption.

LSI circuits contain 100 or more equivalent gate circuits or networks of a similar complexity. LSI circuits are larger functional circuits or are the equivalent of multiple MSI circuits. An LSI circuit often forms a complete system or instrument. The major application of LSI circuits is in semiconductor memories which store binary data. However, there are many different types of complex LSI circuits including electronic calculators, computers and certain types of test instruments.

VLSI circuits usually contain a thousand or more equivalent gate circuits. Chips with this much circuitry can form complete systems. Extraordinarily large logic networks like these in super mainframe computers can be implemented with VLSI circuits.

Integrated Circuit Packaging

A primary consideration to the integrated circuit user is the packaging and physical characteristics of integrated circuits. There are three basic methods of packaging the silicon chip. These are the TO5 can, the flat pack, and the dual in-line package. These three basic types of packages are illustrated in Figure 4-4.

TO5. The earliest form of package used for integrated circuits was the TO5 can. This is a standard configuration for packaging transistors. This same package was modified by including additional leads. This type of package is still available for some integrated circuits today, but it is not the most popular form. Its advantage is that it can dissipate a substantial amount of heat. For that reason, this package is used mostly with linear integrated circuits.

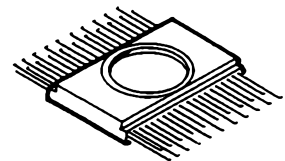
Flat Pack. The flat pack was another type of housing used in the early development stages of integrated circuits. It is the smallest of all available integrated circuit packages and is designed for high density packaging. The packages are flat and are designed to be soldered or spot welded to a circuit board. Circuits can be placed close together and, therefore, a considerable amount of circuitry can be packaged in an extremely small area. Because of their ability to be packed so densely, these integrated circuits are generally made of a ceramic material that can withstand high temperatures. Closely packaged circuits cause heating and cooling problems. Therefore, the circuitry must be able to withstand such an environment. Flat pack circuits are used primarily in critical-size applications such as avionics, high reliability military systems and special industrial equipment.

DIP. The most widely used form of integrated circuit packaging is the dual in-line package (DIP). It is slightly larger than the other types available, but it offers many advantages. Such circuits are easy to mount and use. They are designed to be adaptable to machine insertion on printed circuit boards. They are available in various sizes, all the way from an 8-pin package (mini DIP) to a 64-pin package. Most SSI circuits are housed in 8, 14, or 16-pin dual in-line packages. MSI circuits are found in 14, 16, 18, 20, 22, and 24-pin dual in-line packages. LSI circuits, because of their greater size complexity, require a greater number of input and output leads and, therefore, are usually housed in 24, 28, 40, and 64-pin packages.

TO5



FLAT PACK



DIPS

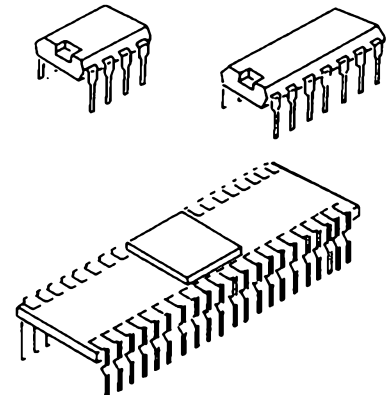


Figure 4-4

Illustrations of typical TO5, flat pack and DIP ICs.

Several different types of dual in-line package materials are used. The most commonly used and least expensive is a plastic package. In this type of package, the integrated circuit chip is spot welded to a metal lead frame. The entire circuit is then encapsulated by an injection molded plastic technique.

For some critical integrated circuits, several types of ceramic packages are used. These are capable of withstanding higher temperatures and are generally hermetically sealed to provide an extra clean and safe environment for the circuit.

Temperature Ranges

Most integrated circuits are rated according to the range of temperatures over which they can operate satisfactorily. Most manufacturers generally specify both a military grade and a commercial or industrial grade circuit. The military grade circuits can be packaged in TO5 cans, ceramic flat packs, or ceramic dual in-line packages. These devices are capable of operating over a wide temperature range, usually from -55°C to $+125^{\circ}\text{C}$. Circuits that perform properly over this wide temperature range are generally much more expensive. Such circuits are used only in high quality military equipment or in industrial equipment that is to be operated in severe environments.

For most general applications the commercial or industrial grade integrated circuits can be used. These are generally housed in plastic packages and are capable of operating over the 0°C to 70°C temperature range. Other temperature ranges are sometimes specified for different types of integrated circuits. Check the manufacturer's data sheet for specific information on temperature ranges.

Self Test Review

7. Most digital ICs are:
 - a. Thin film.
 - b. Thick film.
 - c. Hybrid.
 - d. Monolithic.
8. The IC containing the most gates is:
 - a. SSI
 - b. MSI
 - c. LSI
 - d. VLSI
9. A functional digital IC containing 50 gates is classified as:
 - a. SSI
 - b. MSI
 - c. LSI
 - d. VLSI
10. The most popular IC package is the:
 - a. TO5 can
 - b. flat pack
 - c. DIP
11. The two types of DIP packaging materials are _____ and _____.
12. List the two temperature ranges of most digital ICs.
Military _____
Commercial/Industrial _____

Answers

- 7. (d) Monolithic.
- 8. (d) VLSI
- 9. (b) MSI
- 10. (c) DIP
- 11. Plastic, ceramic
- 12. Military: -55°C to $+125^{\circ}\text{C}$
Commercial/Industrial: 0°C to $+70^{\circ}\text{C}$

TRANSISTOR TRANSISTOR LOGIC

There are many different types of integrated circuit logic elements used in implementing digital equipment. All of them perform the basic logic functions but have different characteristics, capabilities and limitations. Different types of digital logic circuits have been developed to meet special needs. Over the years a variety of circuits have emerged.

One of the biggest and most important design decisions made by an engineer designing digital equipment is in the selection of a type of digital logic circuit. In this section we are going to discuss the most popular form of bipolar integrated circuit logic elements, transistor transistor logic. Non-saturating bipolar circuits and MOS digital integrated circuits will be considered in following sections.

The most popular and most widely used type of digital bipolar IC is transistor transistor logic (TTL or T^2L pronounced T squared L). Its popularity is primarily the result of its extremely low cost and the availability of a wide variety of SSI logic elements and MSI functional circuits. Its ease of use, high performance characteristics and interfacing capability are other features that make it desirable. A number of special types of TTL circuits are available to match special needs. Even though TTL integrated circuits have been available since the early 1960's, this type of logic circuit continues to remain popular for use in new equipment designs. Most of the experiments you will perform in this program use TTL integrated circuits.

Circuit Operation

Figure 4-5 shows the circuit of a typical TTL logic gate. It operates from a single +5-volt power supply and has typical logic levels of .4 volts for binary 0 (low) and 3.6-volts for a binary 1 (high). The circuit consists of three basic sections: a multiple emitter input transistor (Q1), a phasesplitter transistor (Q2), and a totem-pole output circuit consisting of transistors Q3 and Q4. The multiple emitter-base junctions of transistor Q1, along with R1, form a diode gate. The primary advantage of this arrangement over individual diodes is that higher speed operation can be obtained.

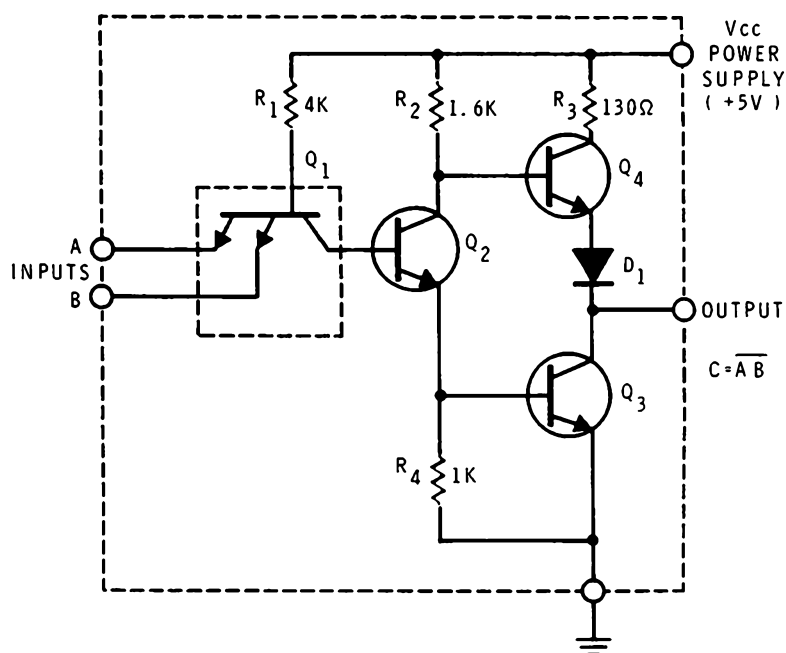


Figure 4-5

A typical transistor-transistor logic gate.

The phase splitter transistor (Q2) is a circuit that provides complementary drive signals for the two output transistors. The output circuit consists of transistors Q3 and Q4. These transistors are stacked one upon the other. Thus, this arrangement is given the name totem-pole. Transistor Q3 is simply a shunt transistor switch. Q4 in this circuit essentially serves as an active load resistor for Q3. In some logic circuits, the collector of the output transistor is returned through a collector resistor to the supply voltage. This collector resistor is known as a pull-up resistor because it causes the output to be pulled up to the supply voltage when the output transistor cuts off. In the TTL gate, Q4 serves as an active *pull-up*

resistor. Current to any shunt load on the output is supplied by this transistor. This arrangement provides a much lower output impedance in the high output state; and, therefore, higher speed operation can be obtained. In logic circuits using a pull-up resistor, any shunt output capacitance must be charged through the collector pull-up resistor. This charging time can be long depending on the amount of shunt capacitance and the value of the collector resistor. With the active pull-up arrangement in the TTL gate, any output capacitance can be charged more quickly through the very low impedance represented by Q4.

To simplify the discussion of the TTL logic circuit, it is convenient to show a diode equivalent of the key parts to this circuit. This diode equivalent arrangement is shown in Figure 4-6. Diodes D1 and D2 in this circuit represent the emitter-base input junctions of transistor Q1. Diode D3 represents the base-collector junction of transistor Q1. Diode D4 represents the emitter-base junction of Q2, and D5 is the emitter-base junction of Q3. Study the diode equivalent in Figure 4-6 and relate it to Figure 4-5. Keep in mind that a PN junction silicon diode requires approximately 0.7 volts across it before it conducts. The forward voltage drop across this diode is also approximately 0.7 volts. Since diodes D3, D4 and D5 are connected in series, the voltage at point X will be the sum of the individual voltage drops or in this case approximately $3 \times 0.7 = 2.1$ volts. A voltage less than 2.1 volts at point X will mean that all three diodes will be cut off.

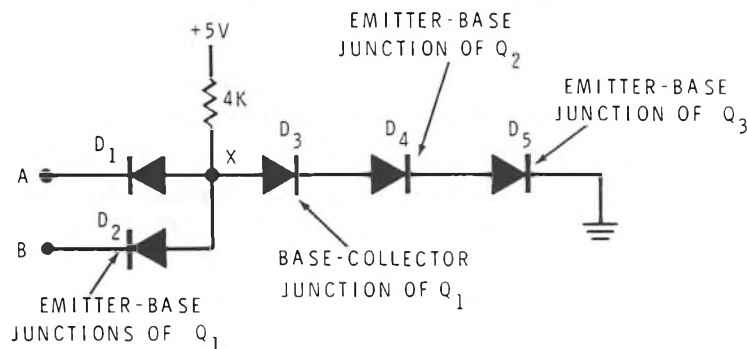


Figure 4-6

Diode equivalent circuit of TTL gate.

Now consider the operation of the circuit. If either one or both of the logic inputs are at their binary 0 level, 0.4 volts or less, the associated input emitter junction will conduct. The voltage at the base of Q1, (point X in Figure 4-6) will be the input logic level voltage plus the drop across the input emitter-base diode, in this case approximately $0.4 + 0.7 = 1.1$ volts. Current will flow through the input diode whose input is low and through the 4K pull-up resistor. Since the voltage at the base of Q1 (point X in Figure 4-6) is less than that required to cause the three diode string

to conduct, the base-collector junction of Q1 will not conduct. The emitter-base junctions of Q2 and Q3 will not conduct therefore these transistors will be cut off. With Q2 off, base current will be supplied to transistor Q4 through resistor R2. Q4 will conduct if an output load is connected to ground. The output voltage at this time will be the supply voltage $+V_{CC}$ less the drop across diode D1, Q4 and the 130 ohm collector resistor. A typical TTL binary 1 output voltage level is approximately $+2.4$ to $+3.6$ volts. A binary 0 voltage level at either or both of the inputs will produce a binary 1 output.

Now assume binary 1 logic levels applied to both inputs. A typical binary 1 logic level input will be $+2.4$ volts or higher. Most of the inputs will be driven from other TTL output circuits and, therefore, the output voltage will in most practical situations approach $+3.6$ volts. The diode equivalent string D3, D4, and D5 in Figure 4-6 will conduct through the 4K resistor. The voltage at point X in Figure 4-6 will be about $+2.1$ volts. Therefore, with $+3.6$ volt inputs, D1 and D2 will be reverse-biased, thus cut off.

The emitter-base junctions of Q2 and Q3 will be forward-biased as well as the base-collector junction of Q1. With Q2 conducting, its collector voltage is lower than that required to turn Q4 on. Base current normally supplied to Q4 through R2 is shunted away by the conduction of Q2. With Q2 conducting Q3 will saturate. At this time the output voltage is the emitter-collector saturation voltage of Q3 which will be $+0.4$ volts or less. As you can see, with binary 1's on both inputs, the output will be binary 0. From this circuit description you can see that the circuit performs the NAND function for positive logic and the NOR function for negative logic. The truth tables in Figure 4-7 sum up the operation of the basic TTL gate, while the typical characteristics are summarized in Table I. Notice that figures B and C are NOT equivalent circuits.

INPUTS		OUTPUT
A	B	C
$+ .4V$	$+ .4V$	$+3.6V$
$+ .4V$	$+3.6V$	$+3.6V$
$+3.6V$	$+ .4V$	$+3.6V$
$+3.6V$	$+3.6V$	$+ .4V$

A

INPUTS		OUTPUT
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

B

INPUTS		OUTPUT
A	B	C
1	1	0
1	0	0
0	1	0
0	0	1

C

Figure 4-7

Truth tables for typical TTL logic gate (A) electrical, (B) positive logic NAND, (C) negative logic NOR.

Table I

TTL Characteristics

Type of logic: Current sinking

Propagation delay: 1-40 nanoseconds

Power dissipation: 1-25 milliwatts

Fan out: 10-30

Noise immunity: high

Logic levels: binary 0 = + .4 volts

binary 1 = +3.6 volts

Basic gate form: positive NAND/negative NOR

Supply Voltage V_{CC} + 5 volts \pm 10 percent

TTL integrated circuits continue to be one of the most popular and widely used forms of logic elements. Many new equipment designs continue to use this type of circuit. Many manufacturers supply TTL circuits and new circuits are developed regularly. The wide range of SSI and MSI types make TTL circuitry perhaps the most versatile line of digital integrated circuits available. The most common type of TTL circuits are the 7400 series originally developed by Texas Instruments. Almost all other integrated circuit manufacturers second source this series of TTL circuits. Other TTL circuits are also available. These include the 9300 series made by Fairchild and the 8000 series manufactured by Signetics. All of these types of TTL circuits are compatible with one another. In this program you will use and demonstrate many different types of the 7400 series of TTL ICs.

Special TTL Variations

All TTL integrated circuits whether SSI or MSI, combinational or sequential use the basic TTL gate circuit shown in Figure 4-5. In addition, there are several other versions of this TTL circuit made for special applications. These include gates for low power operation, higher speed operation or special logic functions.

Low Power TTL. Low power TTL circuits are similar to the basic TTL circuit described earlier. The only difference is that the resistor values in the circuit are approximately ten times higher, meaning that the power consumption of the circuit is one tenth of that of the standard circuit. Low power TTL circuits are excellent for applications requiring a versatile high speed logic line with minimum power consumption. Increasing the values of the internal resistances, causes the propagation delay of the circuit to be increased. The propagation delay in a typical low power gate is approximately 30 to 40 nanoseconds. High speed is sacrificed for low power consumption.

High Power TTL. The high power TTL circuit is basically the same as the standard circuit considered earlier. In this circuit resistor values are decreased significantly in order to improve operating speed. Typically, the gate propagation delay is reduced to approximately 6 nanoseconds. This increase in speed is accompanied by a power dissipation approximately twice that of a standard gate. This is approximately 22 milliwatts average power dissipation per gate. High power TTL circuits have for the most part been replaced by the newer Schottky TTL circuits which are not only faster, but also consume less power.

Schottky TTL. The transistors in a TTL logic circuit operate in the saturation mode. To achieve higher speed operation than that obtainable with a standard or high power TTL gate, non-saturating transistors must be used. This is what is done to improve the speed of operation of Schottky TTL circuits.

The circuit of a Schottky TTL gate is basically the same as the standard TTL gate circuit we discussed earlier. The primary difference is that a diode is connected between the base and collector of each transistor in order to prevent those transistors from saturating. See Figure 4-8A.

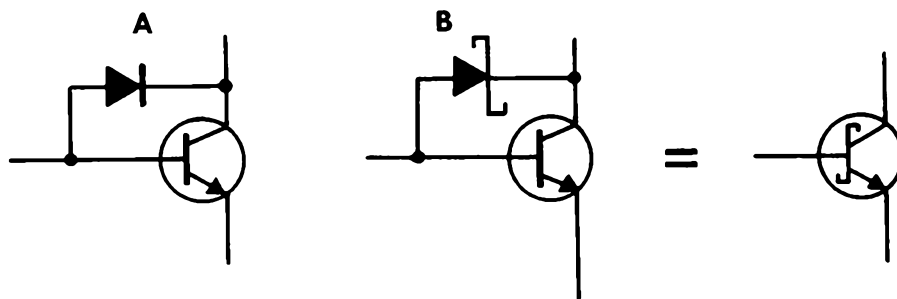


Figure 4-8

Hot carrier diode clamped transistors used in Schottky TTL circuits to prevent saturation and increase switching speed.

When the transistor begins to turn on, its collector voltage will drop quickly to some low value. When it drops beyond a certain point, the diode will conduct and shunt current away from the collector-base junction that would normally conduct during saturation. The diode effectively clamps the collector to a voltage sufficiently high to keep the base-collector junction reverse biased. This circuit permits the condition of saturation to be closely approached but still avoided.

The diode used to provide the clamping that prevents saturation is a hot carrier or Schottky diode. This type of diode, unlike other semiconductor diodes, is not a PN junction type. Instead, a Schottky diode is the junction of a metal such as gold or aluminum, and N-type semiconductor material. These diodes are not separate units. Instead, they are part of the complex integrated circuit diffusion on the silicon chip. These diodes are high speed in operation because they do not have the normal charged storage normally associated with PN junction diodes. The forward voltage drop or voltage required for the diode to conduct is also much less than a standard PN junction.

The Schottky clamp diode as it is used on the transistors in a TTL gate is illustrated in Figure 4-8B. Note the special symbol used to represent the Schottky diode. The special symbol on the right in this figure is a Schottky diode clamped transistor. You will see this symbol used in schematic diagrams of Schottky TTL logic circuits.

The primary advantage of the Schottky TTL gate is its higher speed of operation. Since the transistors do not saturate, no charge storage problems occur. Gate propagation times as low as 3 nanoseconds are possible with this type of circuit. At the same time, Schottky TTL gates achieve this rate of switching at a power dissipation of approximately 19 milliwatts, something less than the high power TTL circuit. Advanced Schottky TTL circuits with propagation delays as low as 1 nanosecond with a power dissipation of 2 milliwatts are now being made. Special low power Schottky TTL circuits with a propagation delay of 10 nanoseconds and a power dissipation of 2 milliwatts are also available. This form of TTL has one of the most favorable speed-power trade-offs of any digital integrated circuit.

Three-State TTL and Data Busses. Three-state TTL integrated circuits are a special version of TTL circuits whose output can assume three states instead of the normal two. Besides the binary 0 and binary 1 logic levels normally associated with a TTL gate output, the three-state circuit has a third *open* state. This open state represents a very high impedance and is essentially equivalent to disconnecting the TTL totem pole output circuit from the output pin on the integrated circuit. This particular type of circuit is useful in digital systems using multiplexed or bussed data transmission.

A data bus is a group of wires, transmission lines or cables over which digital information or binary numbers are transferred in parallel from one point to another. There are basically two types of data busses, unidirectional and bi-directional. On a unidirectional bus, data is transferred in only one direction. On a bi-directional bus, data can move in either direction. Most digital busses are bi-directional in nature.

Instead of having multiple parallel paths for the transmission of digital data in two directions, a common bus is used and the information is transferred from one place to another on a time shared basis. While data from one particular source is being transferred, other data waits until the current transfer is complete. The concept of having one bus serve as a carrier for multiple signals is known as multiplexing. Circuits not currently in use are disabled, while those sending and receiving data on the bus are activated.

Figure 4-9 shows a simplified diagram of a typical bi-directional digital data bus. Only one of several identical bus lines is shown. This bus line is generally responsible for transmitting one bit of data of a multiple-bit binary word. The bus itself may be simply a cable several feet long or a transmission line several hundred feet long. Gates 1, 2, or 3 can transmit one bit of information down the bus line to be received by gate 8. Only one of the three gates will be enabled at a time to transmit the desired data. Note that the same bus can be used to transmit binary information from either gate 6 or gate 7 down the transmission line to be received by gates 4 or 5. Just keep in mind that only a single data transmission may take place at any given time, but that it may be in either direction from any one of several sources or to several destinations.

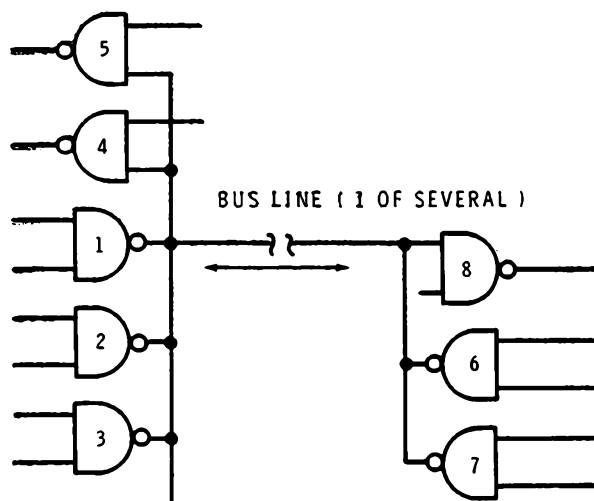


Figure 4-9

Bi-directional digital data bus.

The digital bus is relatively easy to implement with logic circuits using collector pull-up resistors. The outputs of the gates from which the digital data comes are simply connected in parallel as illustrated in Figure 4-10. By connecting their outputs directly together, we effectively parallel the collector resistors, thereby reducing the total resistance to one half the value of an individual resistor. The two output transistors then share a common collector resistance. With this arrangement either transistor Q_1 or Q_2 can bring the output to the binary 0 condition. If Q_1 conducts and Q_2 is cut-off or if Q_2 conducts and Q_1 is cut-off, the output will be binary 0. The only time that the output will rise to $+V_{CC}$ is when both Q_1 and Q_2 are cut-off. The way digital data is transmitted by one gate then is to disable the gates not responsible for transmitting data. This is done by applying the appropriate input to the gate so that its output transistor is cut-off. This permits the other transistor to control the state of the output.

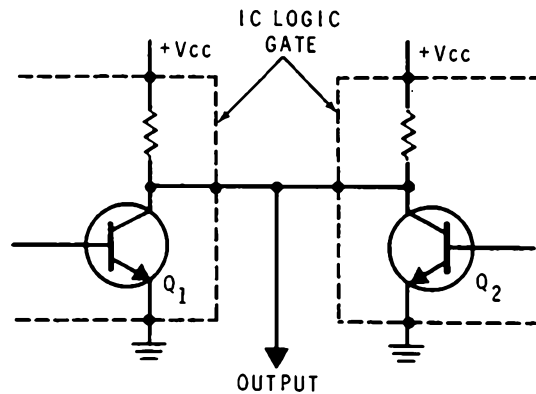


Figure 4-10

Paralleling gate outputs
to share a common output line.

Parallel gate outputs can form what is known as the wired AND connection. It is given this name simply because either transistor Q_1 OR Q_2 can bring the output to the binary 0 level. This connection is frequently used to implement the logical AND function without the need of an additional gate.

Because of the active pull-up circuit in the totem pole output of a TTL gate, TTL circuits cannot be wired in the AND arrangement. Improper operation or damage can occur. For this reason TTL ICs cannot be used in bussing operations. To overcome this problem, open collector TTL circuits can be used. In these circuits, the active pull-up stage is eliminated and the collector of the shunt output transistor is made available at an output pin. To this is connected an external collector pull-up resistor. The wired-AND arrangement can then be used. However, because the active pull-up transistor is removed, one of the primary advantages of a TTL gate is eliminated. The active pull-up produces higher speed operation and lower output impedance, both of which are desirable not only from a speed standpoint but from one of improving noise immunity.

The disadvantage of not being able to use TTL circuits in bus applications was overcome by the development of three-state logic. This type of logic was originally introduced by National Semiconductor Corporation as Tri-State Logic. Three-state logic is a special form of TTL that retains the basic TTL circuit configuration including the totem pole active pull-up output circuit. However, additional circuitry has been added to produce an optional high impedance third state that can effectively remove from any common bus line those circuits not transmitting data.

A typical three-state TTL circuit is shown in Figure 4-11A. The circuit arrangement is basically identical to the TTL gate circuit we discussed earlier. Q_1 is the multiple emitter input transistor, Q_2 is the phase splitter while Q_3 and Q_4 form the totem pole output circuit. Transistor Q_3 has been added in order to provide better control of output transistor Q_4 . Together, Q_3 and Q_4 form a compound or Darlington transistor with high gain. Components D_1 , Q_6 , Q_7 and Q_8 have been added to control the third state.

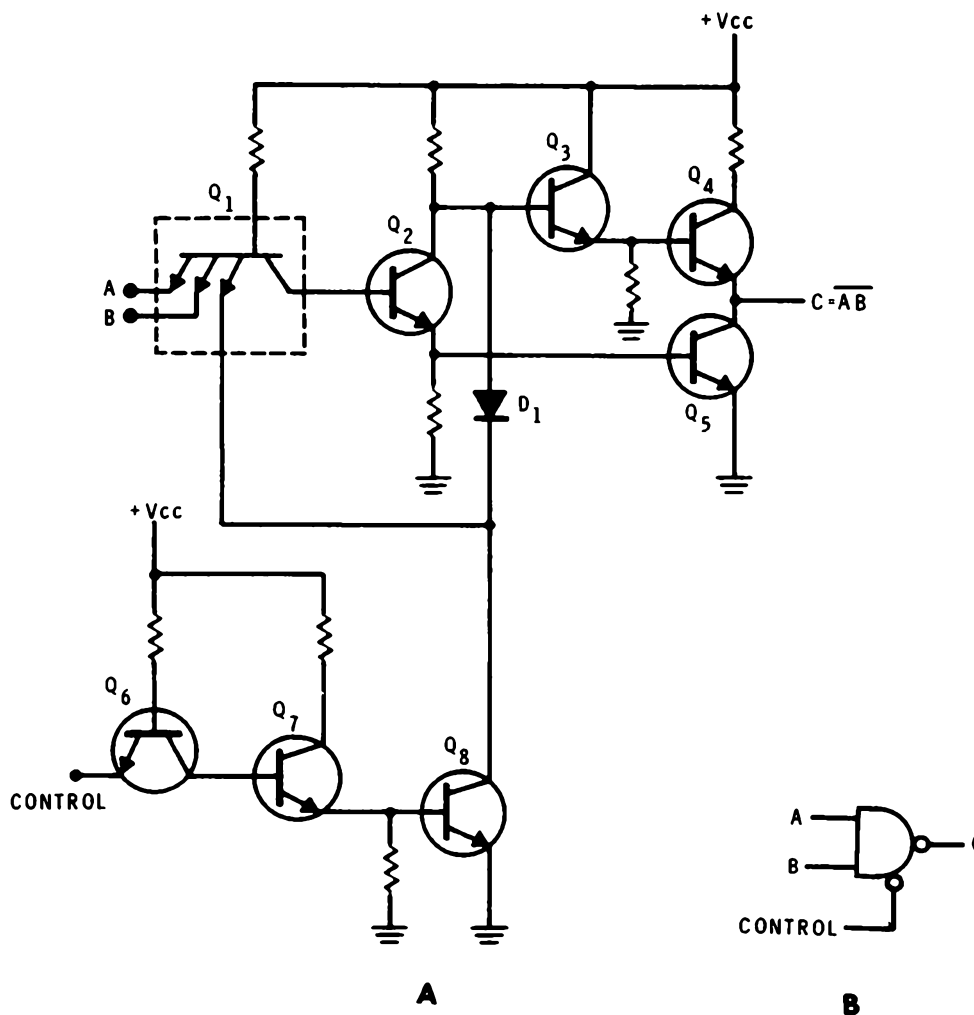


Figure 4-11

Three-state TTL gate schematic (A) and the logic symbol (B).

Whenever the control input is low, transistor Q_6 saturates. When it saturates it causes the collector of Q_6 and hence the base of Q_7 to be nearly the same low input value. This means that Q_7 and Q_8 are cut-off. The TTL gate then functions normally. Here the standard TTL output logic levels for binary 0 and binary 1 are achieved according to the input states.

When the control input is made a binary 1, Q_6 cuts off. The emitter-base junctions of Q_7 and Q_8 become forward biased through the base-collector junction of Q_6 and the associated base resistor. With Q_8 saturated, the third input to Q_1 is brought to ground along with the cathode of diode D_1 . As you recall, when any one or more of the inputs of a TTL gate are brought to a binary 0 level, the output is forced high. This is done by turning on the active pull-up transistor Q_4 and turning off the shunt output transistor Q_5 . But in this case, both output transistors Q_4 and Q_5 are cut-off. When Q_8 saturates, the cathode end of D_1 is grounded. All the base current for Q_3 is shunted away. This causes Q_3 and Q_4 to cut-off. With both output transistors Q_4 and Q_5 cut-off, the output is effectively an open circuit. Looking from the output of the gate back into the circuit, any load sees an extremely high impedance. Because of the high quality of the circuit and the low leakage, any load sees essentially an open circuit. With this arrangement, any number of three-state TTL outputs may be paralleled to form a common bus line. When data is to be transmitted, all of those gates not transmitting data will have their control lines at binary 1 so that their outputs represent an open circuit. Only the gate designated to transmit data will be enabled.

Figure 4-11B shows the logic symbol normally used to represent a three-state TTL gate.

Self Test Review

13. A standard TTL gate performs what logic function for positive logic?
 - a. AND
 - b. OR
 - c. NAND
 - d. NOR
14. If all inputs of a TTL gate are binary 1, the output will be
 - a. binary 0
 - b. binary 1
 - c. indeterminate
15. The typical TTL logic levels are
 - binary 0 . _____ volts
 - binary 1 . _____ volts
16. Two features that make the TTL gate faster than other types of gates are _____ and _____.
17. Schottky TTL is faster than standard TTL because
 - a. smaller resistor values are used.
 - b. it consumes more power.
 - c. hot carrier diodes are faster than regular diodes.
 - d. non-saturating transistors are used.
18. Three-state TTL has three possible output states. These are _____, _____ and _____.
19. The type of TTL gate that can be wire ANDed is _____.
20. For TTL loads, a TTL circuit acts as a
 - a. current source
 - b. current sink.

Answers

13. (c) NAND
14. (a) binary 0.
15. binary 0 = +0.4 volts
binary 1 = +3.6 volts
16. multiple emitter input transistor, totem pole or active output circuit.
17. (d) non-saturating transistors are used.
18. binary 0, binary 1, open
19. open collector
20. (b) current sink for TTL loads.

EMITTER-COUPLED LOGIC

The major switching speed limitation of bipolar logic circuits is the charge storage that occurs in the base region when both the emitter-base and base-collector junctions of a bipolar transistor are forward-biased to achieve saturation. The time required for this charge to be eliminated delays the turn-off of the transistor. Switching speeds can be greatly increased if this delay time is minimized or eliminated. In Schottky TTL circuits this storage problem is eliminated by preventing the transistors from saturating. This significantly increases the switching speed. Non-saturating transistor circuits offer the best potential for fast logic.

Another major form of logic circuits using non-saturating bipolar transistors is called emitter coupled logic (ECL). Also known as current mode logic, this circuitry is essentially a differential amplifier configuration which effectively prevents transistor saturation. ECL ICs are the highest speed logic circuits available today.

Circuit Operation

Figure 4-12 shows the schematic diagram of a typical ECL logic gate.

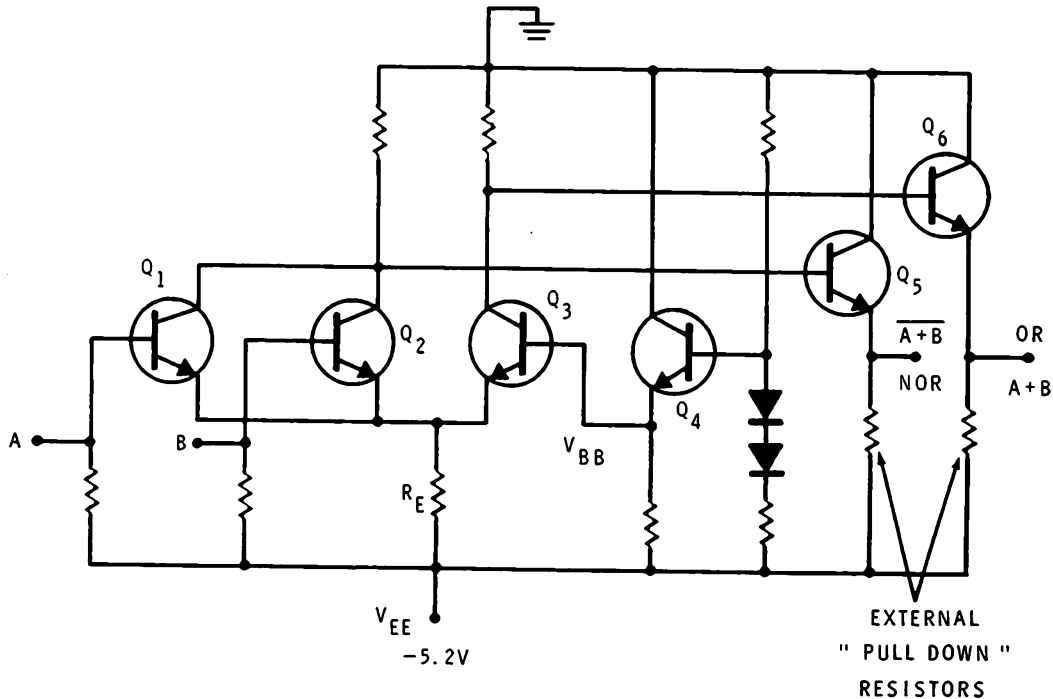


Figure 4-12

Typical emitter-coupled logic gate.

Transistors Q_1 and Q_2 along with Q_3 form a differential amplifier. The inputs (A and B) are applied to transistors Q_1 and Q_2 whose emitter and collector connections are in parallel to form one side of the differential amplifier circuit. If additional inputs are required, more transistors are paralleled. Q_3 is the other side of the differential amplifier. Input logic levels are typically -1.75 volts (binary 0) and $-.9$ volts (binary 1). The output and input voltage swing is typically the difference between these two voltage levels or $(1.75 - .9) = .85$ volts. The supply voltage V_{EE} is -5.2 volts. The supply voltage V_{EE} and R_E form a current source that supplies a fixed current whose level is below the point that permits saturation.

In Figure 4-12, transistor Q_4 and the associated components form a temperature stabilized voltage source that biases the base of Q_3 to approximately -1.3 volts. The emitter of Q_3 will be approximately $.8$ volts more negative than its base because of its emitter base voltage drop. Therefore, the voltage at the emitters of Q_1 , Q_2 , and Q_3 will be approximately $(-1.3 - .8) = -2.1$ volts.

Assume that both logic inputs A and B are at the binary 0 logic level of -1.75 volts. With this condition, transistors, Q_1 and Q_2 will not conduct because the emitter-base bias is insufficient. At this time the collectors of Q_1 and Q_2 are high while the collector of Q_3 is low. These two output levels are buffered by output emitter followers Q_5 and Q_6 and produce the proper binary 0 and binary 1 logic levels. Note that both the normal and complement outputs are available simultaneously.

If any one or both of these logic inputs rise to the binary 1 state ($-.9$ volts), the associated input transistor will conduct. The emitter of Q_1 , Q_2 , and Q_3 will then be one emitter-base junction drop more negative than the input voltage or approximately -1.7 volts. This means that when one or more of the inputs rises to the binary 1 level, the common emitter point will rise from -2.1 volts to -1.7 volts. This voltage will cause transistor Q_3 to cut-off. As you can see the current supplied by the emitter supply voltage and the common emitter resistor R_E switches from Q_3 to the conducting input transistor or transistors. With this arrangement the collectors of Q_1 and Q_2 will be low while the collector of Q_3 will be high. These logic output levels are buffered by the emitter followers Q_5 and Q_6 . The circuit performs the OR and NOR functions for positive logic level assignments.

One of the major advantages of most commercial ECL IC logic circuits is the availability of both the normal and complement outputs simultaneously. This permits both OR and NOR functions to be obtained at the same time. No external inverters are required. Note that the emitter follower "pulldown" resistors are usually external to the integrated circuit itself. This permits the resistors to be located remotely at the end of a transmission line or in another desirable location depending upon the exact circuit configuration and application. It is also possible to tie together the OR or NOR outputs of ECL logic circuits to permit the wired OR function.

Circuit Characteristics

ECL logic circuits are extremely versatile, easy to use and produce high quality results. However, these circuits are normally higher in cost and of course consume significantly more power than other types of logic circuits. Their only real advantage is their high speed. Naturally, this high speed capability should be used only where absolutely necessary. Other types of logic are preferred for slower speed applications. The most widely used form of ECL is Motorola's MECL and 10K series.

Table II

ECL Characteristics

Type of logic: unsaturated, current sourcing

Propagation delay: .5–3 nanoseconds

Power Dissipation: 40–60 milliwatts

Fan Out: 10–25

Noise Immunity: High

Logic levels: binary 0 = –1.75 volts

binary 1 = –.9 volts

Basic gate form: OR/NOR

Supply voltage V_{EE} : –5.2 volts

Self Test Review

21. The basic ECL circuit is a(n)
 - a. inverter
 - b. differential amplifier
 - c. saturated switch
 - d. emitter follower
22. ECL gate outputs can be wire ORed.
 - a. True
 - b. False
23. For positive logic in an ECL gate
 - binary 0 = _____ volts
 - binary 1 = _____ volts
24. An ECL gate performs what functions in negative logic.
 - a. AND
 - b. OR
 - c. NAND
 - d. NOR
25. With ECL loads, ECL gates are
 - a. current sources.
 - b. current sinks.

Answers

21. (b) differential amplifier
22. (a) True
23. binary 0 = - 1.75 volts
binary 1 = - .9 volts
24. (a) AND
(c) NAND
25. (a) current sources

METAL OXIDE SEMICONDUCTOR INTEGRATED CIRCUITS

Metal oxide semiconductor field effect transistors (MOSFETs) or insulated gate field effect transistors offer numerous advantages over bipolar transistors in digital circuits. First, these devices are simpler in construction and therefore can be made much smaller. Because they occupy less space, higher density logic networks can be placed on a given silicon chip. This permits large scale digital integrated circuits to be readily constructed. Another advantage of the MOSFET is its high impedance and therefore, low power consumption. MOS digital integrated circuits consume only a fraction of the power of equivalent bipolar circuits.

The big disadvantage of MOS digital integrated circuits is their lower speed. The high impedance and capacitive nature of these circuits produce switching speeds that are several orders of magnitude slower than bipolar digital integrated circuits. Despite their low speed nature, these circuits have nevertheless found wide application in those areas not requiring high speed operation. Recent technological advances in the manufacturing of MOSFET circuitry have also helped their switching speed approach that of some bipolar circuits.

There are two basic types of MOSFETs used in MOS digital ICs: the P-channel and the N-channel. As indicated earlier, the MOSFETs used in digital ICs operate in the enhancement mode. In this mode of operation, the transistor is normally cut-off. When the appropriate voltage level is applied between the source and the gate, the transistor suddenly switches on. These devices are near perfect switches in that they have an extremely high impedance when not conducting and an extremely low impedance when conducting.

The earliest MOS devices in use in digital circuits were PMOS circuits. P-type MOSFETs are the simplest and easiest to manufacture. N-channel MOS devices are more difficult to make, however, recent technological advances have simplified their construction. N-channel MOSFETs are smaller in size and switch at higher speeds. Their switching threshold is also much lower making them more compatible with bipolar digital integrated circuits. Another class of MOS logic circuits combines both P- and N-channel devices in a single circuit. These digital circuits are known as complementary MOS or CMOS.

MOS digital integrated circuits are clearly the digital ICs of the future. Their small size, low power consumption and simplicity make them attractive for many medium and large scale applications. Most MOS ICs are LSI. Entire test instruments and computers can be made on a single silicon chip with MOS techniques. Improvements in manufacturing techniques will gradually reduce the switching speeds to levels acceptable to all but the most demanding high speed applications. It is expected that MOS ICs will eventually replace many of the various types of bipolar circuits.

PMOS and NMOS Circuits

When a digital IC circuit is constructed only with P-channel MOSFETs, it is known as PMOS. If the circuitry uses only N-channel enhancement mode MOSFETs, the circuits are referred to as NMOS. The basic circuit configurations are the same for either type of transistor.

Figure 4-13 shows the circuit for an N-channel MOS logic inverter. Q_2 is the standard shunt inverter switch. Transistor Q_1 is connected to form a drain or load resistance. A standard integrated resistance occupies substantially more space than the MOSFET and therefore, if used in any quantity, greatly reduces the amount of digital circuitry that can be placed on a given size silicon chip. The gate and drain of Q_1 are connected together which biases the transistor into conduction. It acts as a resistor.

In operation, this circuit performs like any other inverter. When the logic input voltage is less than the gate-source threshold voltage, Q_2 is cut off. Since the gate and drain of Q_1 are connected together, Q_1 conducts and the output voltage is the supply voltage V_{DD} less the source-drain voltage. When the input voltage exceeds the gate-source threshold of Q_2 (usually about 1.5 volts for most NMOS), Q_2 conducts. Its output voltage drops to a very low level. The on resistance of transistor Q_2 is made significantly less than the resistance of Q_1 by a ratio of at least 20 to 1.

A logic inverter circuit using P-channel MOSFETs is shown in Figure 4-14. This circuit is similar to the N-channel inverter discussed earlier, however, a different means is used to bias the load transistor into conduction. In this circuit, the gate voltage is made more negative than the source voltage by the use of another power supply designated $-V_{GG}$. This voltage causes Q_1 to conduct and act as a resistor. Q_2 in this circuit is the shunt inverter switch. Note the power supply polarity difference in this circuit due to the use of P-channel rather than N-channel transistors. The circuit operation is as described before. When the gate-

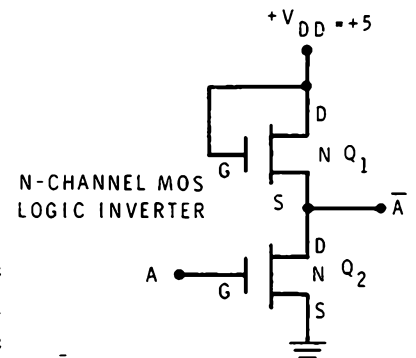


Figure 4-13

N-channel MOS logic inverter.

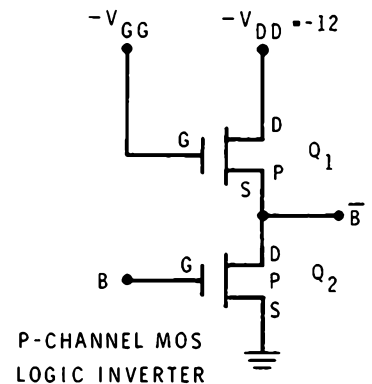


Figure 4-14

P-channel MOS logic inverter.

source threshold voltage is exceeded (4 to 6 volts in most PMOS), transistor Q_2 will conduct and the output voltage will drop to a value near ground. When the gate-source voltage is less than the threshold value, Q_2 will cut off and act as an open circuit. At this time the output voltage is some negative voltage less than $-V_{DD}$. This particular circuit arrangement produces somewhat faster switching speeds than the circuit in Figure 4-13 but the disadvantage is that it requires the additional power supply.

Figure 4-15 shows how the various logic functions are implemented using the MOSFETs. In Figure 4-15A, two N-channel devices are connected in parallel and share a common load (Q_3). If either one or both of the input devices is biased on, the output voltage will drop to a low value. When the input voltages are both less than the threshold voltage, Q_1 and Q_2 will be cut off, allowing the output to rise toward $+V_{DD}$. With this arrangement, the circuit performs a NOR function for positive logic.

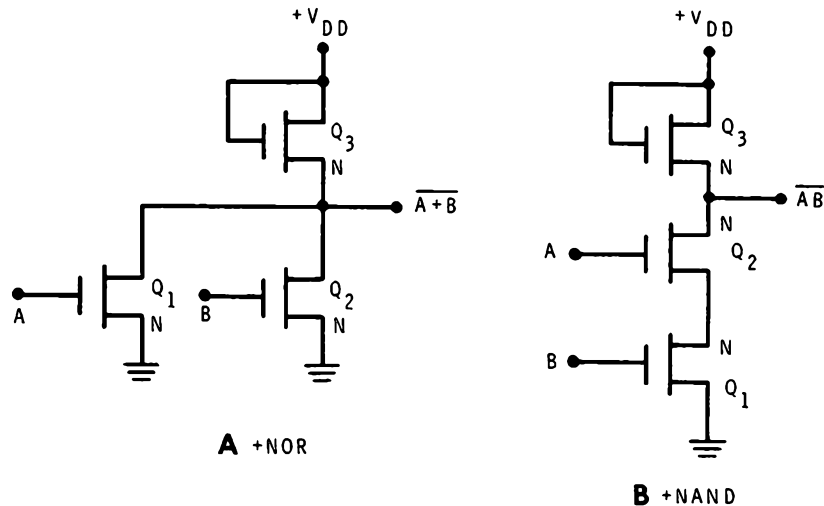


Figure 4-15

NMOS logic circuits.

The circuit in Figure 4-15B performs the NAND function for positive logic. Here transistors Q_1 and Q_2 are connected in series. In order for the output line to be brought low, input signals must appear high at both the A and B inputs simultaneously. If either one or both of the inputs are low, the output will be high. This is the NAND function.

The type of MOS logic circuits that we have discussed here are known as static logic circuits in that they perform logic functions with voltage levels. Another type of MOS logic is also widely used. Known as dynamic logic this circuitry has the same basic configuration as we have discussed here. However, the difference lies in that the circuits take advantage of the capacitive nature of the input to the MOS devices. Here the input capacitors are used to store charges or logic levels temporarily. During operation, high speed clock signals are used to transfer stored charges from one circuit to the next. The advantage of this type of MOS circuitry is still lower power consumption than the static circuits just discussed and higher operating speeds.

Complementary MOS

Complementary MOS or CMOS logic circuits use both P-channel and N-channel enhancement mode MOSFETs. A logic inverter constructed using both types of devices is shown in Figure 4-16. Here the input signal drives the gates of both devices simultaneously. When the input voltage is low or near ground, the gate-source threshold level for Q_2 is less than that required for conduction. Therefore Q_2 is cut off. However with the input low, the gate-source voltage threshold of Q_1 is exceeded. Since the gate is more negative than the source, this P-channel device conducts and connects the supply voltage V_{DD} to the output.

If the input voltage is brought to a high logic level, normally the same as the supply voltage $+V_{DD}$, the gate-source threshold voltage of Q_2 will be exceeded. Q_2 will conduct and act as a very low resistance bringing the output to a low level. At this time with the gate and source of Q_1 approximately at the same levels, the threshold is not exceeded, therefore, Q_1 is cut off. The logic inversion function is performed. It is interesting to note that in this circuit both the current sink and current source modes are used. When Q_2 conducts it sinks current from external loads connected between the outputs and the supply voltage. When Q_1 conducts it supplies current to any load connected between the output and ground. Because of the very low on resistance of a conducting device and the extremely high input impedance of other MOS devices, the logic levels are very nearly equal to ground and the supply voltage $+V_{DD}$.

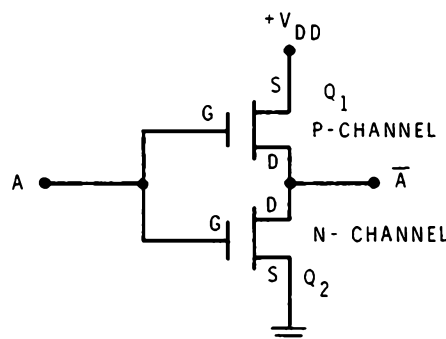


Figure 4-16

A complementary MOS logic inverter.

Figure 4-17 shows a diagram of a typical CMOS logic gate. It consists of two P-channel devices Q_1 and Q_2 connected in series and two N-channel devices Q_3 and Q_4 connected in parallel. When either one or both of the inputs go high to the positive binary 1 voltage level, the associated N-channel transistor Q_3 or Q_4 will conduct. This will cause the output to drop to a voltage level near zero volts indicating a binary 0 output. With either input high, either Q_1 or Q_2 will be cut-off. Since Q_1 and Q_2 are in series, the path between the supply voltage $+V_{DD}$ and the output is not completed unless both Q_1 and Q_2 conduct.

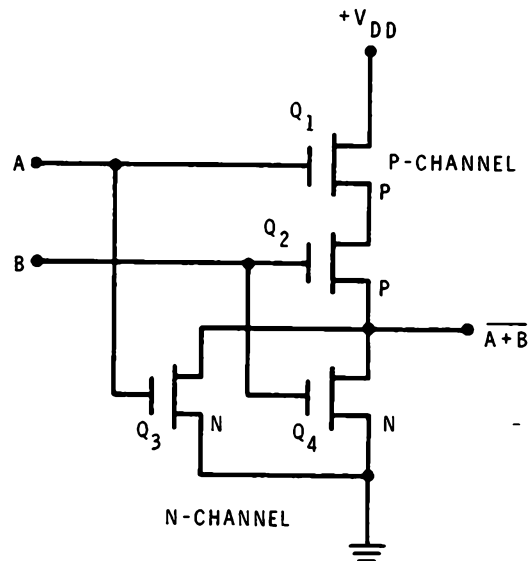


Figure 4-17

CMOS NOR logic gate.

When both inputs go low, Q_3 and Q_4 will become cut-off because their gate-source voltage is below the threshold level necessary to cause conduction. At this time the gate-source voltages of Q_1 and Q_2 will be approximately equal to $+V_{DD}$. Both devices conduct and form a low resistance path between the output and the supply voltage.

Summarizing the operation of this CMOS logic gate, you can see that the NOR logic function is being performed with positive logic. Any logic function including NAND, AND and OR can be performed by suitably arranging the P- and N-channel devices in the circuit.

If a digital designer were to specify the perfect logic circuit for all applications, this logic circuit would have characteristics very similar to those of CMOS. CMOS logic circuits offer a balanced combination of characteristics making it highly versatile and desirable. This type of logic circuit features very low power dissipation, excellent noise immunity, wide power supply voltage variations, high fan out and moderately high speed of operation.

Low power dissipation in CMOS circuits is achieved because there is never a continuous path through any of the devices in the circuit from the supply voltage to ground. A look at the inverter circuit in Figure 4-16 or the NOR gate in Figure 4-17 will indicate this. When the N-channel devices connected between the output and ground are conducting, the P-channel devices between the output and $+V_{DD}$ are cut off. Similarly, when the P-channel device between the supply voltage and the output is on, the N-channel devices from the output to ground are cut off. The current flow that does take place between the supply voltage and ground is that which flows when the output state switches. It is during this time that the P- and N-channel devices may be on together momentarily thereby causing a small current to flow. If the switching rate is high, the rate of occurrence of this current flow increases. The power consumption of a CMOS device increases with its operating frequency because of this effect.

Table III

CMOS Characteristics

Type of logic: Current sinking and current sourcing.

Propagation delay: 10–100 nanoseconds.

Power Dissipation: 10 microwatts (static)
1 milliwatt at 1 MHz

Fan out: 50 +

Noise immunity: very high (45 percent V_{DD})

Logic levels: binary 0 = 0 volts
binary 1 = $+V_{DD}$

Basic gate form: positive NOR/negative NAND

Supply voltage V_{DD} : +3 to +15 volts

The most popular forms of CMOS logic are the 4000 series circuits made by RCA, the 74C series made by National Semiconductor, and the 14000 series manufactured by Motorola. All offer a variety of SSI, MSI and LSI circuits.

Self Test Review

26. P-channel MOS is faster than N-channel MOS.
a. True
b. False
27. MOS combining both P- and N-channel in series is called _____.
28. The primary disadvantage of MOS ICs is _____.
29. The CMOS logic levels are
binary 0 = _____ volts
binary 1 = _____ volts
30. Most MOS ICs are usually
a. SSI
b. MSI
c. LSI
31. The input to a CMOS gate appears primarily as a
a. low resistance
b. high resistance
c. capacitor
d. inductor

Answers

26. (b) False
27. CMOS
28. low speed
29. binary 0 = zero volts
binary 1 = $+V_{DD}$
30. (c) LSI
31. (c) capacitor

INTEGRATED INJECTION LOGIC

One of the newest forms of integrated digital circuits is Integrated Injection Logic, usually abbreviated IIL or I^2L (I squared L). I^2L is a form of bipolar logic used primarily in LSI and VLSI applications. Its very small size permits very high density; and its ultra low power consumption eliminates heat problems and gives long battery life in portable applications. I^2L circuits are also more common than you think. In fact, if you are wearing a digital watch, you have an I^2L chip on you!

Circuit Description

The development of I^2L was an attempt to achieve the speed of operation of bipolar circuits while obtaining the packing density and low power consumption of MOS. The attempt was successful, as VLSI I^2L circuits are widely used today. I^2L designers created the smallest and simplest digital logic circuit available today. A typical I^2L gate occupies only about half the space of even the smallest MOS circuit. The small size is the result of there being essentially only one transistor per gate.

A typical I^2L circuit is shown in Figure 4-18. It consists of Q_1 a multiple collector NPN shunt saturated switch and Q_2 a PNP current source. This circuit is used in various ways to produce the basic logic functions. Q_2 can also be used to supply current to other shunt switches like Q_1 . As shown, the circuit in Figure 4-18 performs as a logic inverter.

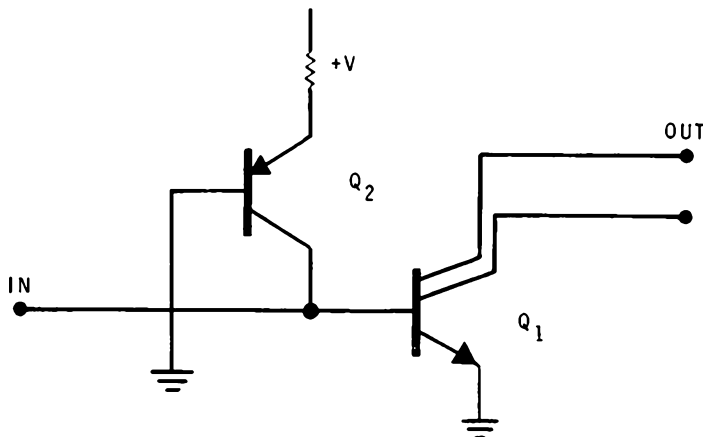


Figure 4-18

Basic I^2L Circuit

To see how the circuit works, refer to Figure 4-19 where two basic I^2L inverters are cascaded. Normal logic levels are approximately $+ .1$ volt for a binary 0 (or near ground) and $+ .7$ volts for a binary 1. Current source Q_4 supplies current to the base of Q_3 . This current is high enough to saturate Q_3 . Therefore, Q_3 conducts and its output voltage $V_{CE}(\text{sat})$ is about $+ .1$ volt or less (binary 0). This condition exists when the input to Q_3 is a binary 1, about $+ .7$ volts. With Q_3 saturated, current from source Q_2 is shunted away from the base of Q_1 . Therefore, Q_1 is cut-off. Its output will, therefore, be $+ .7$ volts or the V_{BE} of the next transistor it drives.

Now assume that the input to Q_3 is brought low ($+ .1$ volt or less). Current from Q_4 is shunted away from the base of Q_3 and Q_3 turns off. Q_2 can now supply base current to Q_1 . Q_1 conducts. The voltage at the collector of Q_3 and base of Q_1 is now $+ .7$ volts, the base-emitter voltage drop V_{BE} of Q_1 . The collector voltage of Q_1 is now $+ .1$ volts or less. As you can see, the two logic levels are derived from the $V_{CE}(\text{sat})$ of the driving transistor (binary 0) and the V_{BE} of the driven transistor (binary 1).

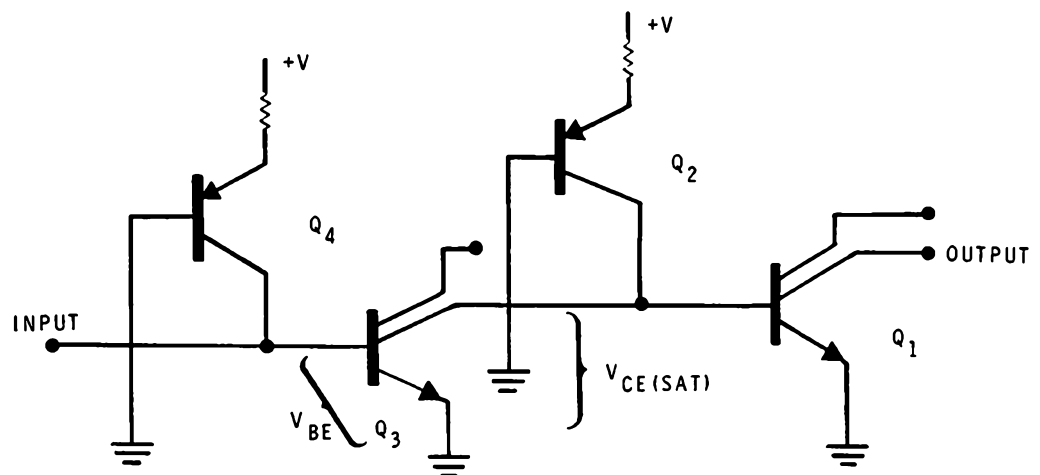


Figure 4-19

Cascaded I^2L Inverters

The basic circuit can be combined in several ways to perform the various logic functions. An I²L NAND gate is illustrated in Figure 4-20A. The only difference between this and the basic circuit is that multiple inputs A and B are connected to the base of Q₁. These inputs are driven by other I²L circuits Q₃ and Q₄. If either one or both inputs are low (0), Q₁ is cut off and the output is high (1). If both inputs are high (1), Q₁ will conduct and its output will be low (0). This is the NAND function as the truth table in Figure 4-20B shows.

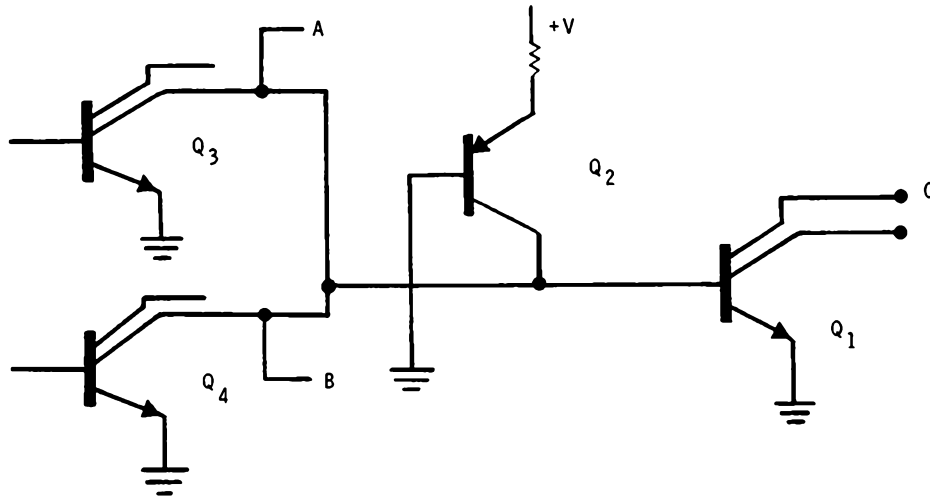


Figure 4-20A

I²L NAND circuit

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Figure 4-20B

Truth Table

An I²L NOR gate is shown in Figure 4-21A. The NOR function is obtained by the wired OR connection. If either or both inputs are high (1); Q₁, Q₃, or both will conduct and the output F will be low (0). When both inputs are low (0), both Q₁ and Q₃ will be cut off and the output will go high (1). The operation is summarized in the truth table of Figure 4-21B. This is the NOR function.

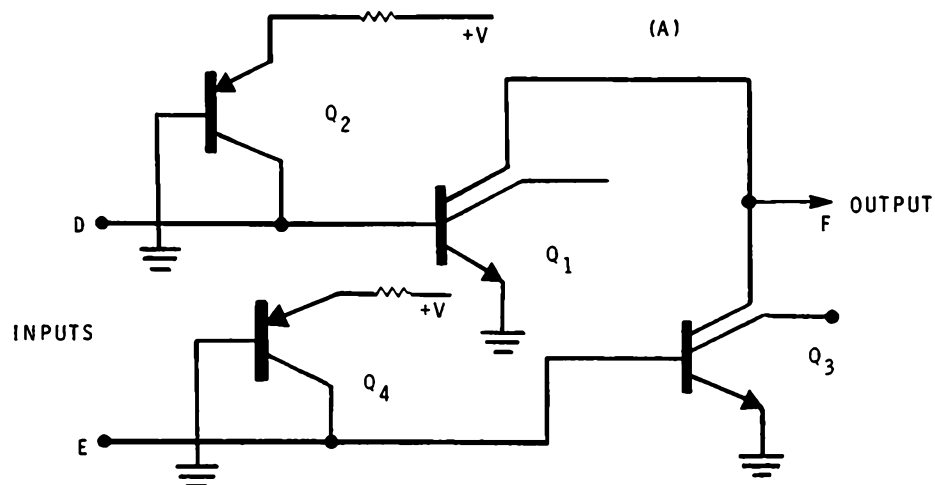


Figure 4-21A

I²L NOR gate circuit

D	E	F
0	0	1
0	1	0
1	0	0
1	1	0

Figure 4-21B

Truth Table

Characteristics and Applications

We have already mentioned the very small size of I^2L circuits. Over 100 I^2L circuits can be constructed in about the same space that one TTL gate occupies. Literally thousands of gates can be made on a single silicon chip. I^2L circuits can achieve the same function of MOS in about one half the space. And, the fewer diffusion and masking steps required in manufacturing I^2L makes for greater yields and lower cost.

As for speed, the typical I^2L gate has a propagation delay of 25 to 50 nanoseconds. Although, depending upon the application, delays can range from 20 to 250 nanoseconds. Keep in mind that this is accomplished with saturated bipolar transistors. Higher speeds are obtainable by using Schottky diodes as with TTL circuits. However, this does increase the gate area and cost.

The power consumption of a typical I^2L gate is in the 6 nanowatt to 70 microwatt range. The dissipation of an average 20 nanosecond gate is 50 microwatts. The power consumption of an I^2L gate is about the same as that of a CMOS gate; however, the I^2L gate is two or three times faster.

As it turns out, I^2L circuits have the most favorable speed-power product of any logic circuit. Typical circuits have a speed-power product of one picojoule or less, which beats even the best CMOS devices.

Finally, the noise margin is fair to good. The simple bipolar circuitry is also easy to interface to almost any other type of IC (TTL, CMOS, etc.). I^2L characteristics are summed up in Table IV.

Another major benefit of I^2L circuits is that the simplicity of their structure permits linear circuits to be easily fabricated on the same chip. With more complex standard digital circuits, it is difficult and not practical economically to put digital and linear circuits on the same chip. But with I^2L , it is easy to incorporate analog circuits such as op amps, comparators, LED/LCD drivers, memory sense amplifiers, and oscillators right on the chip with a complex digital circuit. Today, more and more designs incorporate both linear (analog) and digital functions and, to create complete LSI or VLSI systems, it is desirable to put all circuitry on one chip.

Presently, I²L circuits are used predominantly in digital watches. Their small size and ultra low power consumption make them perfect for the application. There are also I²L memory circuits and micro-processors for special lower power uses. I²L circuits are also used in 35 mm cameras for lighting, exposure, and timing control. Telephone dialing and electrical appliance control circuits have also been developed.

TABLE IV

I²L Characteristics

Type of logic: Current sinking

Propagation delay: 25–50 nanoseconds

Power dissipation: .06 to 70 microwatts

Fan out: 2

Noise immunity: Fair to good

Logic levels: binary 0 = <+.1 volts

binary 1 = +.7 volts

Basic gate form: Positive NAND or NOR

Supply voltage: 1–15 volts

Self Test Review

32. The logic levels of an I²L circuit are:
 binary 0 _____
 binary 1 _____
33. I²L gates are superior to all other types of ICs in:
 a. propagation delay
 b. noise immunity
 c. size
 d. power dissipation
34. I²L circuits are used primarily in _____ and _____ circuits.
35. The main element on an I²L circuit is a:
 a. current source
 b. multi-collector shunt switch
 c. PNP saturated switch
 d. Schottky diode

Answers

32. Binary 0 = +.1 volt
 Binary 1 = +.7 volts
33. c. size
34. LSI, VLSI
35. b. multi-collector shunt switch

SELECTING A DIGITAL INTEGRATED CIRCUIT FOR A SPECIFIC APPLICATION

The most important decision that you will make in designing a piece of digital equipment is in selecting the type of integrated circuit. The success or failure of your design from a performance and economic standpoint will depend directly upon this choice. For this reason, you must carefully consider the requirements of your application.

The first step in selecting a digital integrated circuit is to completely define the system characteristics. Performance, economy, and reliability are the prime requirements. These are specifically stated in the terms of needed system speed, noise immunity, power dissipation and other factors. Once all of the specifications have been detailed, these can be matched against the capabilities and characteristics of the various types of integrated circuits available.

The three key factors in defining a digital system and selecting a type of integrated circuit are speed (propagation delay), power consumption and noise immunity. While these are the most important characteristics in terms of performance and economy, there are other considerations. These include cost, availability, trends and the need for complex functions. The cost and availability factors are obvious. Lowest cost circuits are best if they meet all of the requirements of your application. And the circuits you choose must be readily available. It is desirable to select a circuit with several sources of supply.

Trends

The semiconductor industry is one of the most volatile in the field of electronics. New technological developments occur frequently and therefore some devices are quickly obsoleted while others with improved characteristics are added. These changes take place so quickly that it is difficult for any designer to select an integrated circuit with long life and a stable price. It is important for you to follow the trends by reading the manufacturer's literature and the technical publications. By staying abreast of the latest technology you will be better prepared to select a circuit that not only meets your design requirements but also will have favorable life, cost and availability trends.

Complex Functions

A highly desirable characteristic of any integrated circuit type is the availability of medium scale integrated circuits (MSI). These are functional circuits either of the combinational or sequential type. Since most digital equipment is made up of only a few basic types of circuits (decoders, counters, etc.), a large portion of the equipment can be designed by simply interconnecting the functional circuits. It is much more economi-

cal to use MSI functions than to implement these same functions with SSI gate circuits. By using MSI functions the design time is reduced substantially. In addition the total package count and size is proportionally reduced. Savings in power consumption and in assembly labor can be significant. The more MSI functions available in a digital integrated circuit logic line, the more advantageous it is.

Trade Offs

Keep in mind that all of these factors are somewhat inter-related. Your choice of a type of digital IC will be a compromise based on your application and the circuits available. The speed-power trade off is one of the most critical trade-offs in selecting a circuit. Noise immunity is also a factor that may have to be traded off with some other characteristic depending upon the types of circuits available. You may have to juggle circuit specifications with cost and availability.

Figure 4-22 shows the primary characteristics of all of the types of digital logic circuits discussed in this section. It will permit you to compare the different types for your application. Only these types should be considered for new product design. TTL, ECL, and CMOS are available in a variety of SSI and MSI devices. MOS and IIL are used in LSI and VLSI only.

Comparison Of IC Logic Family Characteristics					
Characteristic	TTL	ECL	MOS	CMOS	IIL
Fan-out	10	25	20	50 +	2
Cost	low	medium to high	medium to high	low to medium	low to medium
Power dissipation per gate (mW)	12 to 22	40 to 60	0.2 to 10	0.01 static 1 at 1 MHz	<.07
Noise generation	high	low to medium	medium	low medium	low
Immunity to external noise	good	good	good	very good	fair to good
Temperature range (°C)	-55 to +125 0 to 70	-55 to +125	-55 to +125 0 to 70	-55 to 125 -40 to 85	0 to 70
Typical supply voltage (V)	+5	-5.2	-12 (PMOS) +5 (NMOS)	+1.5 to 18	+1 +.15
Avg. propagation delay per gate (ns)	1 to 40	.5 to 3	300 (PMOS) 50 (NMOS)	70	20-50
Avg. clock rate (MHz)	15 to 120	200 to 1000	2 (PMOS) 5 to 10 (NMOS)	5 to 10	1-10

Figure 4-22

Figure 4-23 shows a chart of speed vs. power consumption for the various types of integrated circuits. Here propagation delay is plotted as a function of power dissipation for all of the most popular types of digital logic circuits. Naturally, the best circuit is the one with the shortest propagation delay and the least amount of power dissipation. You can see from this chart the type of circuit with the most favorable speed-power rating is low and advanced power Schottky TTL.

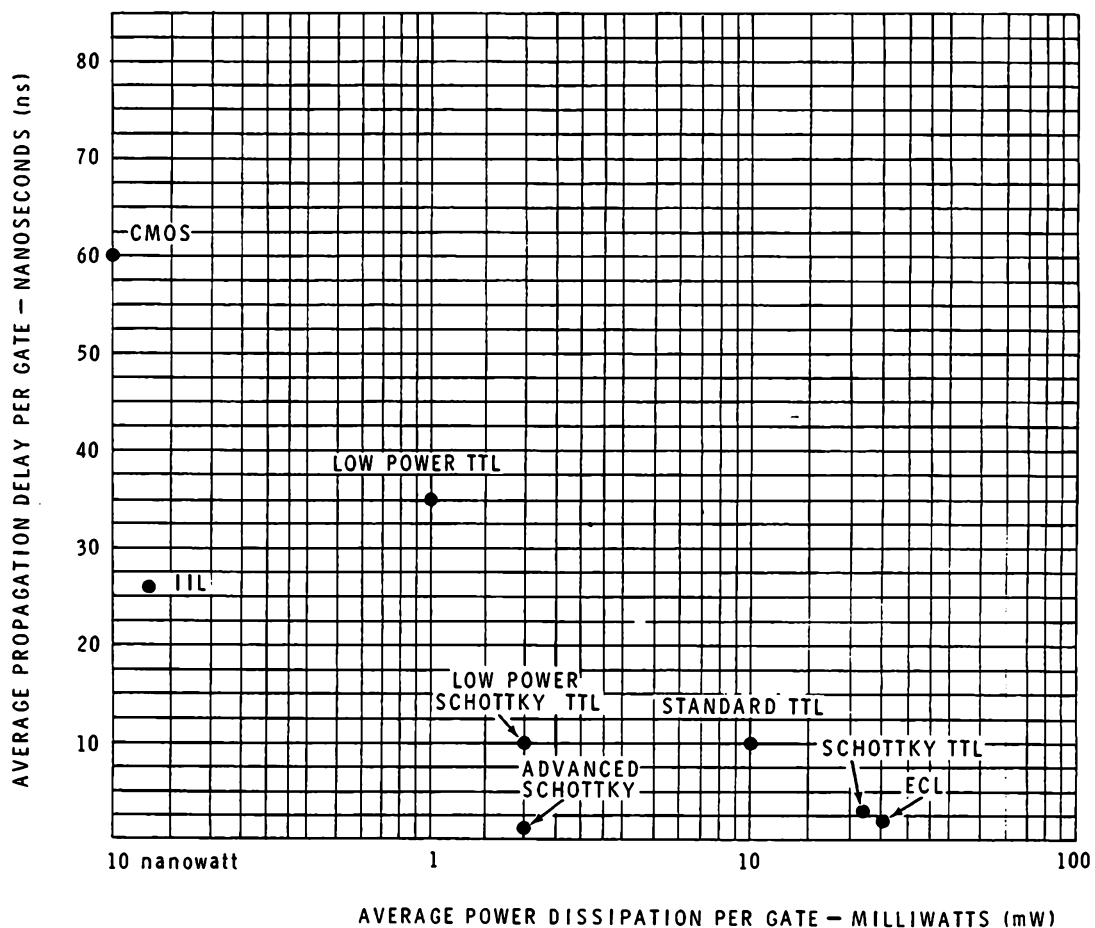


Figure 4-23

Speed (propagation delay) vs.
Power dissipation for popular
digital integrated circuits.

Self Test Review

36. The digital ICs primarily recommended for new equipment design are _____, _____, _____, _____ and _____.
37. Name three *non-technical* characteristics that affect a designer's choice of a digital IC.
- a. _____
 - b. _____
 - c. _____
38. The most important trade-off in selecting a digital IC is the one involving _____ and _____.

Answers

36. TTL, CMOS, ECL, MOS, IIL
37. Cost, availability, trends, complex functions (MSI)
38. speed, power

EXPERIMENT 5

TTL Logic Gates

OBJECTIVE:

To demonstrate the operation and characteristics of a TTL logic gate and to show how it can be used to perform any of the three basic logic functions.

Materials Required

- 1 – SN 74LS00N (7400) quad-two input TTL integrated circuits (443-728)
- 1 – 1N4149 silicon diode (56-56)
- 1 – 560 ohm resistor
- Heathkit ET-3200 Digital Design Experimenter
- DC Voltmeter or logic probe.

Procedure

1. Mount the 7400 TTL integrated circuit on the breadboarding socket. Be sure that it is seated firmly straddling the notch in the socket and that none of the pins are bent. Connect pin 14 to +5 volts and pin 7 to GND to supply power. Figure 4-24 shows the pin connections.

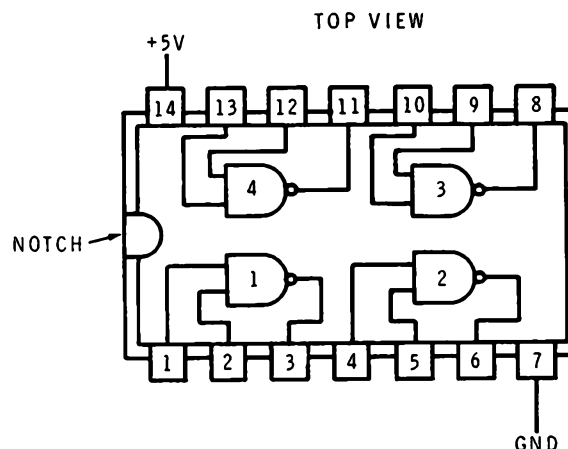


Figure 4-24

Pin connections for 7400 TTL IC.

2. Connect one of the four gates in the IC as shown in Figure 4-25. The input will come from data switch SW1. You will monitor the input and output states with the L1 and L2 LED indicators and your DC voltmeter.

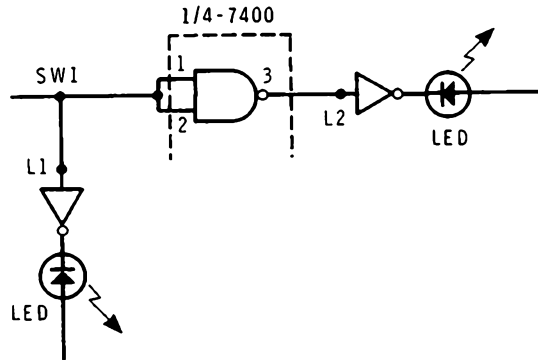


Figure 4-25

3. Set SW1 to the down position then the up position. Measure the DC input (pins 1 and 2) and output (pin 3) voltage for each position. Record your data in Table I. Also note the LED indicator input/output states.

Table I

INPUT	OUTPUT

4. Assuming positive logic, the output logic levels are:
 binary 0 = _____ volts.
 binary 1 = _____ volts.
5. Study Table I. What logic function is being performed?
 _____.
6. Connect the diode-resistor circuit shown in Figure 4-26 to the output of the TTL gate. This circuit simulates a load of about 7 TTL gate inputs. Again measure the input and output voltages of the circuit for both positions of SW1. Record your data in Table II.

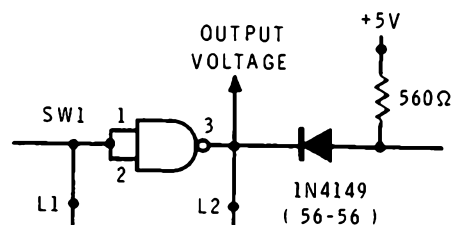


Figure 4-26

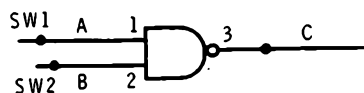


Figure 4-27

Table II

INPUT	OUTPUT

7. Compare the output data in Tables I and II and account for any differences. Does the loading affect the binary 0 or binary 1 output state most? _____.
8. Wire the circuit shown in Figure 4-27. Remove the 1N4149 and 560 Ω resistor load used in the previous steps. The inputs come from SW1 and SW2. You will measure the output voltage C at pin 3 of the 7400 IC.
9. With SW1 and SW2, apply the input voltages given in Table III. Measure and record the output voltage for each set of inputs.

Table III

INPUTS		OUTPUT
A(SW1)	B(SW2)	C
0V	0V	
0V	+5V	
+5V	0V	
+5V	+5V	

10. Using positive logic convert your electrical truth table in Table III into 1's and 0's in Table IV.

Table IV

A	B	C

11. Study Table IV. What logic function is being performed?

12. Using negative logic, convert the data in Table III into 1's and 0's and record in Table V.

Table V

A	B	C

13. Study Table V. What logic function is being performed?

14. Remove the wires connecting pins 1 and 2 of the IC to SW1 and SW2. Let the gate inputs hang free. Note the output state.
With open inputs, the TTL gate output is _____ volts or binary _____ for positive logic. This means that an open input acts like a binary _____.
15. Wire the circuit shown in Figure 4-28. With SW1 (A) and SW2 (B), apply the states shown in Table VI. Record the state for each set of inputs. Observe LED indicators L1, L2, and L3 to obtain your input and output data. Use positive logic (binary 1 = on, binary 0 = off).

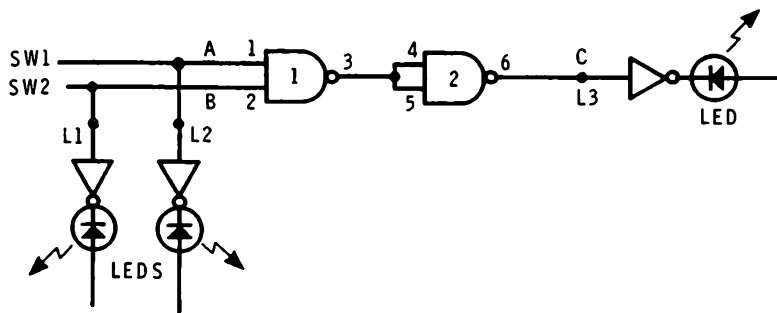


Figure 4-28

Table VI

A	B	C
0	0	
0	1	
1	0	
1	1	

16. Study the circuit in Figure 4-28 and the data in Table VI. What logic function is being performed? _____.
17. Connect the circuit shown in Figure 4-29. Monitor the inputs and output on LED indicators L1, L2, and L3. With SW1 (A) and SW2 (B), apply the input shown in Table VII. Record the output state corresponding to each set of inputs. Use positive logic.

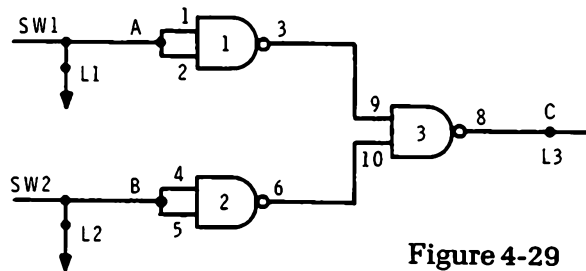


Figure 4-29

Table VII

A	B	C
0	0	
0	1	
1	0	
1	1	

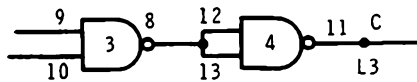


Figure 4-30

18. Study Figure 4-29 and Table VII. What logic function is being performed? _____.
19. Modify your circuit in Figure 4-29 by adding the fourth gate in the 7400 to the output as shown in Figure 4-30. Only the output change is shown. The rest of the circuit stays as in Figure 4-29.
20. Using SW1 (A) and SW2 (B) data switches and monitoring LED indicators L1, L2 and L3, apply the states shown in Table VIII. Record the output state for each set of inputs.

Table VIII

A	B	C
0	0	
0	1	
1	0	
1	1	

21. Study Table VIII. What logic function is being performed?
_____.

Discussion

In Steps 3 through 7 you demonstrated how one of the four gate circuits in the 7400 IC could be used as an inverter. The two inputs are tied together to form a single input line. With this arrangement, the gate performs as a TTL logic inverter where the input and output are complementary.

In Step 4 you measured the input and output voltages. The input from SW1 is 0 and +5 volts. The approximate output levels should have been binary 0 = +0.1 volts and binary 1 = +3.8 volts.

In Step 6 you loaded the TTL circuit with a current sinking type load then observed the output under load conditions. You should have found that with the load, the binary 0 output state was higher by about .1 volt than it was for no load. In the binary 0 output state, the shunt output transistor must sink the load current. The greater the sink current the higher the output voltage ($V_{CE\text{ sat}}$). If the sink or load current is too high, the transistor will come out of saturation and the gate output voltage will be even higher. For this type of TTL gate, the maximum fan out is 10. Each gate input represents a sink current of 1.6 ma. Therefore each TTL gate output can sink 16 ma. If the load is greater, the output voltage in the binary 0 state may rise to more than .8 volts which is the maximum allowable level for a binary 0.

In Steps 9, 10 and 11 you demonstrated the basic logic function of the TTL gate. Using positive logic it performs the NAND function. In Steps 12 and 13 you should have found that the basic TTL gate performs the NOR function with negative logic.

In Step 14 you investigated the effect of open inputs. With both inputs open the output should have been binary 0, this indicates that open inputs act like binary 1 levels.

You demonstrated how TTL gates could be connected to perform the basic AND function in Steps 15 and 16. In Figure 4-28, gate 1 is a NAND while gate 2 is connected as an inverter to complement the NAND output to produce the AND output.

You connected TTL gates as an OR circuit and demonstrated its function in Steps 17 and 18. In Step 19 you added an inverter to the output of the OR to produce the positive NOR function. The AND and OR circuits are commonly used with the TTL gate. However, the NOR version is rarely used. A separate positive NOR TTL logic element is available (SN74LS02N) to eliminate the need to interconnect a 74LS00 as we did here.

EXPERIMENT 6

CMOS Logic Gates

OBJECTIVE:

To demonstrate the operation and characteristics of a CMOS logic gate and to show how it can be used to perform any of the three basic logic functions.

Materials Required

1 – CD4001AE (4001) quad two input CMOS integrated circuit (443-695)

Heathkit ET-3200 Digital Design Experimenter

DC Voltmeter or logic probe

Procedure

1. Mount the 4001 CMOS integrated circuit on the breadboarding socket. Be sure that it is seated firmly straddling the notch in the socket and that none of the pins are bent. Connect pin 14 to +5 volts and pin 7 to GND to supply power. Figure 4-31 shows the pin connections.

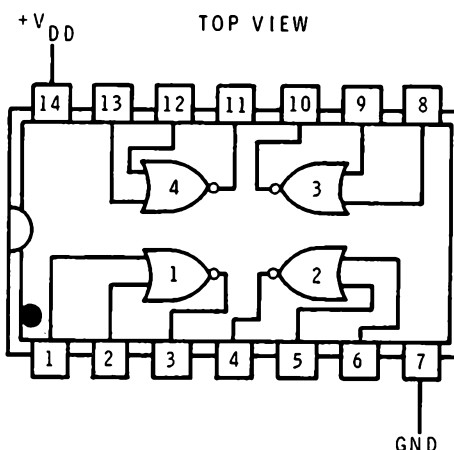


Figure 4-31

Pin connections for 4001 CMOS IC.

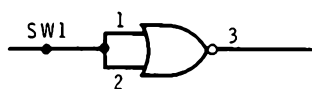


Figure 4-32

2. Connect one of the four gates in the IC as shown in Figure 4-32. The input will come from data switch SW1. You will measure the input and the output states with your DC voltmeter.

3. Set SW1 first to the down position then the up position. Measure the DC input (pins 1 and 2) and output voltage (pin 3) for each position. Record your data in Table I.

Table I

INPUT	OUTPUT

4. Assuming positive logic, the output logic levels are:
 binary 0 = _____ volts.
 binary 1 = _____ volts.
5. Study Table I. What logic function is being performed?

6. Wire the circuit shown in Figure 4-33. The inputs come from SW1 and SW2. You will measure the output voltage C at pin 3 of the 4001 IC.
7. With SW1 and SW2, apply the input voltages given in Table II. Measure and record the output voltage for each set of inputs.

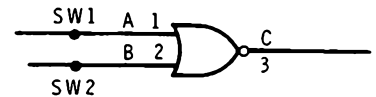


Figure 4-33

Table II

INPUTS		OUTPUT
A (SW1)	B (SW2)	C
0V	0V	
0V	+5V	
+5V	0V	
+5V	+5V	

8. Using positive logic, convert your electrical truth table in Table II into 1's and 0's in Table III.

Table III

A	B	C

9. Study Table III. What logic function is being performed?

10. Using Negative logic, convert the data in Table II into 1's and 0's and record in Table IV.

Table IV

A	B	C

11. Study Table IV. What logic function is being performed?

12. Remove the wire connecting pins 1 and 2 of the IC to SW1 and SW2. Let the gate inputs hang free. Measure the output voltage. With open inputs, the CMOS gate output is _____ volts or binary _____ for positive logic. This means that an open input acts like a binary _____.
13. Wire the circuit shown in Figure 4-34. With SW1 (A) and SW2 (B), apply the input states shown in Table V. Record the output state for each set of inputs. Observe LED indicators L1, L2, and L3 to obtain your input and output data. Use positive logic (binary 1 = on, binary 0 = off).

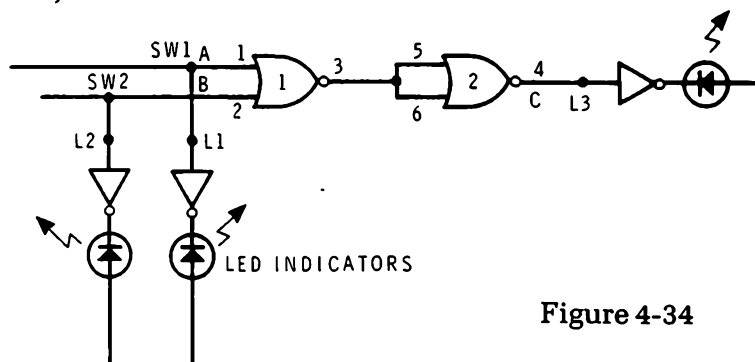


Figure 4-34

Table V

A	B	C
0	0	
0	1	
1	0	
1	1	

14. Study the circuit in Figure 4-34 and the data in Table V. What logic function is being performed? _____.
15. Connect the circuit shown in Figure 4-35. Monitor the inputs and output on LED indicators L1, L2, and L3. With SW1 (A) and SW2 (B), apply the inputs shown in Table VI. Record the output state (C) corresponding to each set of inputs. Use positive logic.

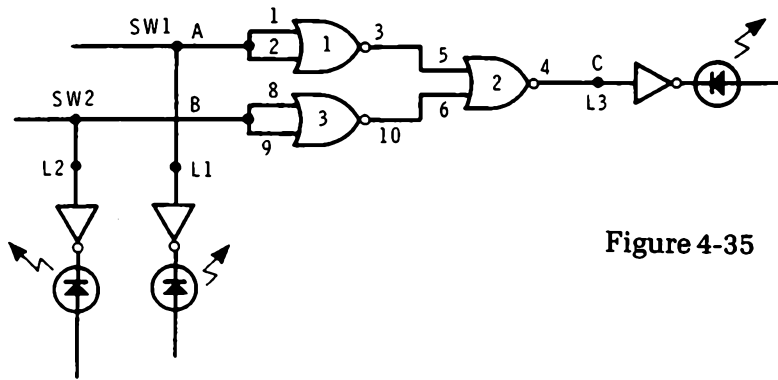


Figure 4-35

Table VI

A	B	C
0	0	
0	1	
1	0	
1	1	

16. Study Figure 4-35 and Table VI. What logic function is being performed? _____.
17. Modify your circuit in Figure 4-35 by adding the fourth gate in the 4001 to the output as shown in Figure 4-36. Only the output change is shown. The rest of the circuit stays as in Figure 4-35.

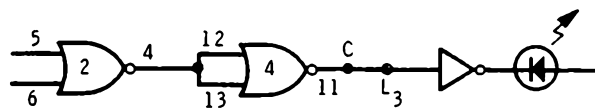


Figure 4-36

18. Using SW1 (A) and SW2 (B) data switches and monitoring LED indicators L1, L2 and L3, apply the states shown in Table VII. Record the output state (C) for each set of inputs.

Table VII

A	B	C
0	0	
0	1	
1	0	
1	1	

19. Study Table VII. What logic function is being performed?
_____.

Discussion

In Steps 3 through 5 you demonstrated how one of the four gate circuits in the 4001 1C could be used as an inverter. The two inputs are tied together to form a single input line. With this arrangement, the gate performs as a logic inverter where the input and output are complementary.

The input from SW1 is 0 and +5 volts. The approximate output levels should have been binary 0 = +.1 volts and binary 1 = +5 volts.

In Steps 7, 8 and 9 you demonstrated the basic logic function of the CMOS gate. Using positive logic it performs the NOR function. In Steps 10 and 11 you should have found that the basic CMOS gate performs the NAND function with negative logic.

In Step 12 you investigated the effect of open inputs. With both inputs open, the output could have been any voltage between 0 and +5 volts. Open inputs can result in operation in the linear region due to biasing by noise at the high impedance inputs. CMOS inputs should never be left open. Unused inputs should be connected to another input or, in a NOR gate, to ground.

You demonstrated that the CMOS gate can be connected to perform the basic OR function in Steps 13 and 14. In Figure 4-34, gate 1 is a positive NOR (negative NAND) and gate 2 is connected as an inverter to complement the NOR and produce an OR output.

You next connected CMOS gates as an AND circuit and demonstrated its function in Steps 15 and 16. In Steps 17, 18, and 19 you added an inverter to the output of the AND to produce the positive NAND function. The AND and OR circuits are commonly implemented with CMOS NOR gates. However, the NAND version shown here is rarely used. A separate positive CMOS NAND logic element is available (CD4011AE) to eliminate the need to interconnect a 4001 as we did here.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this Unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and answer all of the questions first before checking the answers.

1. The two major classes of digital circuits are:
 - a. bipolar
 - b. linear
 - c. hybrid
 - d. MOS
2. The most commonly used IC package for digital ICs is the:
 - a. TO5 can
 - b. Flat pack
 - c. DIP, plastic
 - d. DIP, ceramic
3. Which of the following is **not** a major characteristic of a digital IC?
 - a. size
 - b. power consumption
 - c. speed
 - d. noise immunity
4. The speed of a logic circuit is expressed as:
 - a. power consumption
 - b. propagation delay
 - c. noise immunity
 - d. logic levels
5. Which of the following are current sinking logic?
 - a. MOS
 - b. TTL
 - c. I^2L
 - d. ECL
6. Which of the following are current sourcing logic?
 - a. MOS
 - b. TTL
 - c. I^2L
 - d. ECL

7. Which of the following is **not** a bipolar logic circuit?
 - a. CMOS
 - b. TTL
 - c. I^2L
 - d. ECL
8. The fastest digital circuit is:
 - a. CMOS
 - b. Schottky TTL
 - c. NMOS
 - d. ECL
9. The second fastest digital circuit is:
 - a. CMOS
 - b. Schottky TTL
 - c. NMOS
 - d. ECL
10. The two digital circuits most desirable for battery operation are:
 - a. CMOS
 - b. NMOS
 - c. PMOS
 - d. I^2L
11. The time that it takes a logic circuit to respond to an input and generate an output is called:
 - a. speed-power product
 - b. power dissipation
 - c. propagation delay
 - d. clock frequency
12. Which bipolar circuits are non-saturating?
 - a. ECL
 - b. Standard TTL
 - c. I^2L
 - d. Schottky TTL
13. Which IC type would be most suitable for noisy industrial applications?
 - a. TTL
 - b. ECL
 - c. CMOS
 - d. I^2L

14. Which digital circuit is best based on a comparison of speed-power products?
 - a. 2 picojoules
 - b. 15 picojoules
15. A multiwire connection between digital circuits is usually called a:
 - a. ribbon cable
 - b. bus
 - c. multiplexed line
 - d. wire wrap
16. The time-sharing of one line with multiple signals is called:
 - a. time spacing
 - b. simultaneous transmission
 - c. bidirectional
 - d. multiplexing
17. Time-sharing of multiple lines carrying digital signals is accomplished by using:
 - a. three-state TTL
 - b. wired AND connections
 - c. disconnect switches
 - d. filters
18. When a three-state TTL circuit is in its third state, its output looks like a(n):
 - a. short circuit
 - b. binary 0
 - c. binary 1
 - d. open circuit
19. The ability of a logic circuit to reject unwanted signals is expressed as:
 - a. noise margin
 - b. propagation delay
 - c. power consumption
 - d. logic levels

20. The most widely used type of digital circuit is:
- a. CMOS
 - b. MOS
 - c. TTL
 - d. ECL
21. The speed of a logic circuit is generally proportional to:
- a. power dissipation
 - b. fan out
 - c. noise immunity
 - d. size
22. Four TTL gates with an average propagation delay of 5 nanoseconds are cascaded. The propagation delay of the combination in nanoseconds is:
- a. 4
 - b. 5
 - c. 9
 - d. 20
23. A logic gate has a fan out of 8. It is driving three gate inputs. How many more gate inputs can it drive?
- a. 3
 - b. 5
 - c. 11
 - d. 24
24. The main element in an NMOS logic circuit is a/an:
- a. enhancement mode MOSFET
 - b. depletion mode MOSFET
 - c. saturated bipolar transistor
 - d. Schottky transistor
25. An IC with 50 gates would be classified as:
- a. SSI
 - b. MSI
 - c. LSI
 - d. VLSI

EXAMINATION ANSWERS

1. a.—bipolar, d. MOS
2. c.—DIP, plastic
3. a.—size
4. b.—propagation delay
5. b.—TTL, c. I²L
6. d.—ECL
7. a.—CMOS
8. d.—ECL
9. b.—Schottky TTL
10. a.—CMOS, d. I²L
11. c.—propagation delay
12. a.—ECL, d. Schottky TTL
13. c.—CMOS
14. a.—2 picojoules
15. b.—bus
16. d.—multiplexing
17. a.—three-state TTL, b. wired AND connections
18. d.—open circuit
19. a.—noise margin
20. c.—TTL
21. a.—power dissipation
22. d.—20 nanoseconds
4 levels x 5 nanoseconds = 20 nanoseconds
23. b.—5 Fan out = $8 - 3 = 5$
24. a.—enhancement mode MOSFET
25. b.—MSI

Unit 5

BOOLEAN ALGEBRA

CONTENTS

Introduction	5-3
Unit Objectives	5-4
Unit Activity Guide	5-5
Relating Digital Logic Circuits and Boolean Equations	5-6
Truth Tables	5-14
Boolean Rules	5-21
DeMorgan's Theorem	5-32
DeMorgan's Theorem (Continued)	5-39
Minimizing Logic Circuits	5-45
Using NAND and NOR Gates	5-51
NOR Logic Equivalent Circuits	5-55
NAND Logic Equivalent Circuits	5-65
Experiment 7 – Applying NAND and NOR Gates	5-75
Experiment 8 – The Wired AND Connection	5-82
Unit Examination	5-91
Examination Answers	5-95
Appendix – Unit 5	
Summary of Boolean Rules	5-99

INTRODUCTION

Boolean algebra is the special language of digital logic circuits. It is a mathematical method of expressing, analyzing, and designing logic circuits. It is similar in many ways to conventional algebra. Boolean algebra is easy to learn and extremely useful. It is almost essential to the proper understanding and application of digital circuits. In this unit, you will learn how to use Boolean algebra. You will see how truth tables help you to design and understand logic circuits. And you will learn how to implement practical digital circuits using integrated circuit logic elements.

The specific things you will learn in this unit are outlined in the Unit Objectives which follow. Follow the individual steps listed in the Unit Activity Guide. Check off each item as you perform it and keep track of your time and progress in the spaces provided.

UNIT OBJECTIVES

When you complete this Unit on Boolean algebra, you will be able to:

1. Define Boolean algebra.
2. Write the Boolean expression corresponding to a given logic circuit.
3. Draw the symbolic logic circuit implementing or corresponding to a given Boolean expression.
4. Write the Boolean expression corresponding to a given truth table.
5. Give an example of each of the two basic types of Boolean expressions (sum-of-products and product-of-sums).
6. Minimize a given logic expression using the various rules of Boolean algebra.
7. Implement a given Boolean expression with either NAND or NOR gates.
8. Write the two versions of DeMorgan's theorem.
9. Write the Boolean expression of logic circuits using the wired AND connection.

UNIT ACTIVITY GUIDE

	Completion Time
<input type="checkbox"/> Read "Relating Digital Logic Circuits and Boolean Algebra."	_____
<input type="checkbox"/> Answer Self-Test Review questions 1-5.	_____
<input type="checkbox"/> Read "Truth Tables"	_____
<input type="checkbox"/> Answer Self-Test Review questions 6-7.	_____
<input type="checkbox"/> Read "Boolean Rules"	_____
<input type="checkbox"/> Read "DeMorgan's Theorem."	_____
<input type="checkbox"/> Answer Self-Test Review questions 8-19.	_____
<input type="checkbox"/> Read "DeMorgan's Theorem (Continued)."	_____
<input type="checkbox"/> Answer Self-Test Review questions 20-26.	_____
<input type="checkbox"/> Read "Minimizing Logic Circuits."	_____
<input type="checkbox"/> Answer Self-Test Review questions 27-31.	_____
<input type="checkbox"/> Read "Using NAND/NOR Gates."	_____
<input type="checkbox"/> Read "NOR Logic Equivalent Circuits."	_____
<input type="checkbox"/> Answer Self-Test Review questions 32-34.	_____
<input type="checkbox"/> Read "NAND Logic Equivalent Circuits."	_____
<input type="checkbox"/> Answer Self-Test Review questions 35-39.	_____
<input type="checkbox"/> Perform Experiment 7.	_____
<input type="checkbox"/> Perform Experiment 8.	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Review the Examination Answers	_____

RELATING DIGITAL LOGIC CIRCUITS AND BOOLEAN EQUATIONS

Boolean algebra is a simplified mathematical system used to deal with binary or two value functions. It permits us to express all of the various logic functions, both simple and complex, in a convenient mathematical format. This system gives us a method of understanding and designing digital logic circuits.

The mathematical expression of logic functions permits a convenient means of analyzing and expressing operations in digital circuits. It also aids greatly in design. The proper application of Boolean algebra usually results in the simplest, least expensive, and most efficient logic circuit design.

Most digital equipment in use today is made with integrated circuits. Boolean algebra is used in designing these devices. The applications of integrated circuits in the design of modern electronic equipment, also involves Boolean algebra, but to a lesser extent. At one time, the engineer or technician designing a digital system had to design not only the logic functions but also the circuits to implement them. Boolean algebra was his primary design tool.

Today, the engineer and technician using and designing digital circuits, finds Boolean algebra most valuable in expressing and analyzing logic circuits. His design job is basically that of choosing and using existing integrated circuits to implement the functions required by the application. Occasionally, Boolean algebra will be used to minimize a function and achieve an efficient design.

Review of Basic Functions

A Boolean expression is an equation that expresses the output of a logic circuit in terms of its inputs. You were introduced to Boolean expressions when you studied basic logic gates. The binary inputs and outputs were expressed as letters of the alphabet, alpha-numeric combinations, abbreviations, or short words called mnemonics. For example, in the AND gate in Figure 5-1, the inputs are A and B and the output is C.

Note that the output C is expressed in terms of the inputs. The dot between the A and B indicates the AND function. The output expression $C = A \cdot B$ is read C equals A AND B. Remember that the inputs and outputs are binary signals which may assume either the binary 0 or binary 1 state.

As another example, the output expression of the AND gate in Figure 5-2 is $X = \bar{C} \cdot D \cdot E$. In most Boolean expressions for the AND function, the dot between each input variable can be eliminated and the expression written as a standard algebraic product like $X = \bar{C}DE$. The AND function is sometimes referred to as the logic product.

Consider the output expression for the AND gate in Figure 5-3, $VL = JRF \cdot LEF \cdot KMD \cdot CME$. In this expression, each input is identified by a 3-letter combination called a mnemonic. The output is a 2-letter mnemonic. The dot between each input not only designates the AND function but also helps to separate or distinguish the inputs from one another. Occasionally you will see parenthesis used to separate the inputs and indicate the logic product or AND function.

$$VL = (JRF) (LEF) (KMD) (CME).$$

In the previous examples, you wrote the output equation of a given gate. Now, to be sure you understand the principles, let's translate from the equation to the gate. Consider the circuit that implements the function $W = (F) (MX) (\bar{G})$. See Figure 5-4.



Figure 5-4



Figure 5-1



Figure 5-2

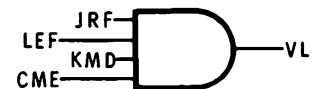


Figure 5-3

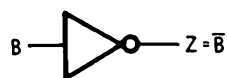


Figure 5-5

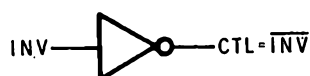


Figure 5-6

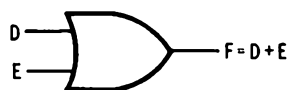


Figure 5-7

Other logic functions like inversion are also expressed with Boolean expressions. For example, if the input to an inverter is B and the output is designated Z , the output expression is $Z = \bar{B}$. The output of an inverter is the complement of the input. This is expressed by putting a bar over the input variable. The term \bar{B} is expressed as NOT B or B NOT. The term NOT refers to logic inversion. See Figure 5-5.

Another example is the logic circuit corresponding to the expression $CTL = \overline{INV}$ in Figure 5-6. The output CTL is the complement of the input INV . $CTL = \overline{INV}$.

Another common logic function is the OR. In an OR gate, the output will be binary 1 if any one or more inputs are binary 1. A typical OR logic gate and its related output expression is shown in Figure 5-7. The plus sign between the input variables designates the OR function. It is sometimes referred to as the logic sum. Figure 5-8 further illustrates the OR function. The output expression of the logic gate in Figure 5-8 is $Y = ZT + \bar{K} + MA$. The plus sign or OR function separates the input variables.

Now, consider the logic circuit for the expression $FN = MAF + \overline{DG} + BF + JS$. See Figure 5-9.

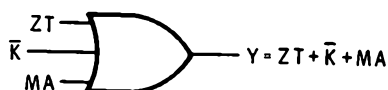


Figure 5-8

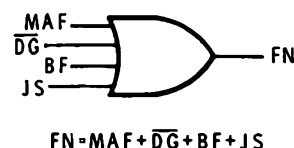


Figure 5-9

Boolean Formats

While there are some simple digital control operations that can be implemented with a single logic gate, more often it is necessary to use a number of logic gates to implement the desired decision-making function. When two or more logic elements are combined, the result is known as a combinational logic circuit. The circuit usually has multiple inputs and either a single output or multiple outputs depending upon its exact function. Any combination of multiple ANDs, ORs, and NOTs is called a combinational logic circuit. Such circuits are used for sophisticated decision-making functions.

There are many common combinational logic circuits used in digital equipment. They perform specific functions that tend to regularly reoccur in digital equipment. However, regardless of the type of combinational logic circuits there are two basic circuit forms. These are referred to as the sum-of-products and product-of-sums circuits. Here the term product refers to the AND function while the sum refers to the OR function.

The AND function is written in the same form as the algebraic product or the multiplication of two variables. ($A \text{ AND } B = AB$) The OR function is written as the sum of two input variables. ($D \text{ OR } E = D + E$). The sum-of-products or product-of-sums expressions combine the AND and OR functions in a variety of ways.

SUM-OF-PRODUCTS

The most commonly used Boolean expression of complex decision-making functions is the sum-of-products. The expression $X = A \cdot B + C \cdot D \cdot E$ is an example.

Figure 5-10 shows the logic circuit implementing this sum-of-products logic function. Here AND gate 1 forms the logic product AB while gate 2 forms the product CDE . These products are summed or logically ORed in gate 3 to form the output expression. This is what is meant by the sum-of-products.

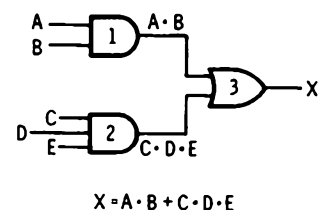


Figure 5-10

Another example of the sum-of-products format is the expression $QT = PJ \cdot \overline{W} + HV \cdot LM + GND \cdot \overline{C} \cdot F$. The various inputs are ANDed together in several combinations which are then ORed together. The circuit for this function is shown in Figure 5-11. The distinguishing feature of this circuit is that the inputs feed AND gates and the output is derived from an OR gate.

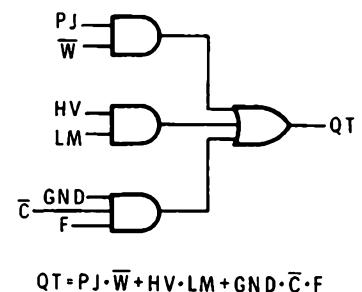


Figure 5-11

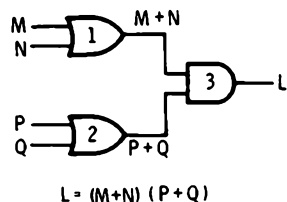


Figure 5-12

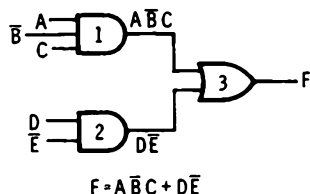


Figure 5-13

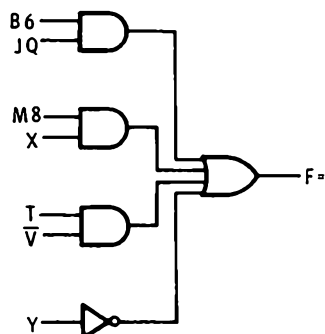


Figure 5-14

PRODUCT-OF-SUMS

The other type of Boolean expression is the product-of-sums. An example is the expression $L = (M + N)(P + Q)$. The equivalent circuit is shown in Figure 5-12. Inputs M and N are ORed together in gate 1 while inputs P and Q are ORed in gate 2. The two logic sums are ANDed in gate 3 to produce the logic product. In this arrangement the distinguishing feature is the OR gate inputs and AND gate output.

Relating Circuits and Equations

You should be able to write the Boolean expression from any logic circuit and draw the logic circuit corresponding to a given Boolean equation. To write the equation of a given logic circuit, you start at the inputs and write the output expression for each gate in the circuit from left to right until the output equation is developed. The unique gate symbols of course will tell you the logic function being performed. The example in Figure 5-13 illustrates this procedure.

You first write the output expressions for the two input AND gates. These expressions become the inputs to the OR gate whose output equation is then written. The result is the Boolean expression for the circuit.

$$F = A\bar{B}C + D\bar{E}$$

Now look at the circuit in Figure 5-14. Its output equation is $F = B6 \cdot JQ + M8 \cdot X + T \cdot \bar{V} + \bar{Y}$. You simply write the output for each of the AND gates and use these as the inputs to the OR gate. Working this way from left to right quickly produces the complete output expression.

This procedure of writing the equation for a given circuit also works on a product-of-sums circuit like the one in Figure 5-15. The output expression of this circuit is $Z = (T + U) \cdot (V + \bar{W}) \cdot X$. The logic sums are formed by gates 1 and 2 and their outputs are combined to form the logic product in the output AND gate 3. Let's take one more example. The output of the circuit in Figure 5-16 is $OPR = (A3 + B4)(JMP + T0 + CLK)$.

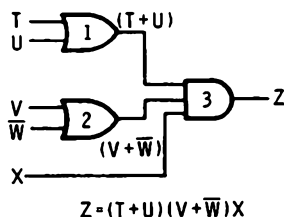


Figure 5-15

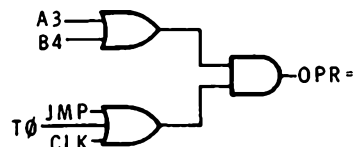
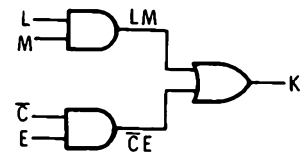


Figure 5-16

To draw the diagram corresponding to a given expression you first study the equation to determine whether it is a sum-of-products or product-of-sums. This will give you the type of output gate. Then you work backward from the output developing the inputs and outputs from right to left.

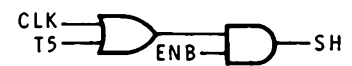
For example, consider the expression $K = LM + \bar{C}E$. This is in sum-of-products form so the output gate is an OR. The inputs to this OR gate are LM and $\bar{C}E$. Each of these is a product that is developed by an AND gate with the appropriate inputs. See Figure 5-17.



$$K = LM + \bar{C}E$$

Figure 5-17

Consider the circuit for the expression $SH = (CLK + T5) ENB$ in Figure 5-18. This expression is in product-of-sums form but here there is only one logic sum ($CLK + T5$). The output is a product, however, and is produced by an AND gate.



$$SH = ENB(CLK + T5)$$

Figure 5-18

One more example is given in Figure 5-19. This is the circuit for the expression $Y = ADD \cdot CY + SUB \cdot \bar{INV} + MPY \cdot ADD \cdot SUB$. This is sum-of-products form. Note that some of the input signals (ADD, SUB) are applied to more than just one of the input gates.

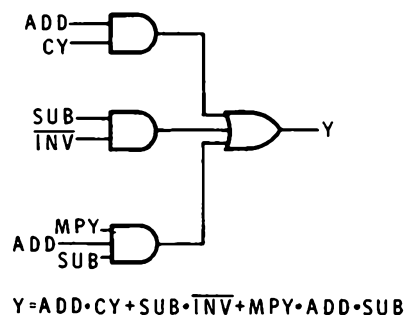


Figure 5-19

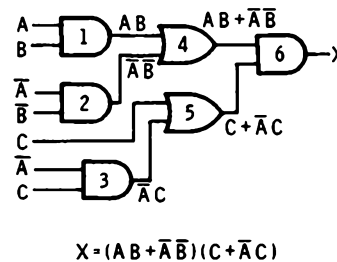


Figure 5-20

So far we have only worked with two levels of logic, that is the inputs are conditioned by two sets or levels of logic gates in cascade: ANDs into OR or ORs into AND. Other more complex logic networks use three, four or even more levels of logic. The expression and circuit in Figure 5-20 is one example. Here both the sum-of-products and product-of-sums formats are combined. There are three levels of logic in this circuit. The input signals must propagate through a series of three gates before a level change occurs at the outputs. Input C on gate 5 only propagates through two levels.

Another 3-level logic circuit is given in Figure 5-21. The output expression is $F = (A + \bar{B})(\bar{A} + B) + (B + \bar{C})\bar{A}$. Again you can see that the sum-of-products and product-of-sums are combined to form a three level logic circuit.

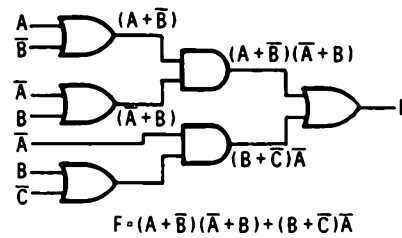


Figure 5-21

Self Test Review

- The most commonly used logic format is:
 - Sum-of-products.
 - Product-of-sums.
 - Combination of A. and B.
 - Neither A. or B.
- Draw the circuit for the expression $M = \bar{V}W + XY + \bar{W}\bar{X}Y$.
- Draw the circuit for the expression $F = T(U + \bar{V})(\bar{T} + W)$.
- Write the Boolean expression for the circuit in Figure 5-22.
- Write the Boolean equation for the circuit in Figure 5-23.

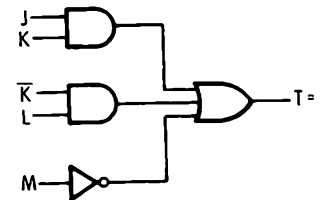


Figure 5-22

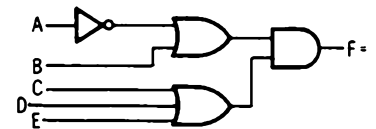


Figure 5-23

Answers

- A. Sum-of-products.
- See Figure 5-24.
- See Figure 5-25.
- $T = JK + \bar{K}L + \bar{M}$
- $F = (\bar{A} + B)(C + D + E)$

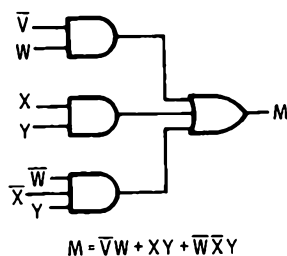


Figure 5-24

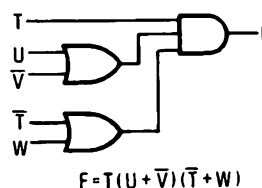


Figure 5-25

TRUTH TABLES

One of the most useful tools for analyzing, designing or otherwise working with digital circuits is the truth table. A truth table is a chart that lists all possible input and output signal combinations for a given logic circuit. The truth table is a tabular listing of all input and output states in binary 0 and 1 form. The truth table completely defines the operation of the circuit. You can look at a truth table and quickly identify which specific combination of input states that will produce a binary 1 (or binary 0) output.

Truth tables are used in defining the operation of simple logic circuits like inverters and gates as well as more complex combinational logic circuits. You have already seen how truth tables are used in defining the basic logic operations such as AND, OR, and NOT.

INPUT A	OUTPUT \bar{A}
0	1
1	0

Figure 5-26A

Figure 5-26 shows the truth tables for the three basic logic functions: invert, AND and OR.

In Figure 5-26A, the truth table for an inverter shows both the inputs and corresponding outputs. The output \bar{A} is always the opposite or complement of the input A.

The truth table for a two input AND gate is illustrated in Figure 5-26B. The inputs A and B define all possible input combinations.

INPUTS		OUTPUT
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Figure 5-26B

Column C shows the output state corresponding to each input combination. The output is binary 1 only when both inputs are binary 1. This is true of any AND gate.

The truth table defining the operation of an OR gate is given in Figure 5-26C. The output is a binary 1 if any or all inputs are binary 1.

INPUTS		OUTPUT
D	E	F
0	0	0
0	1	1
1	0	1
1	1	1

Figure 5-26C

The total number of possible input conditions in a logic circuit is 2^n where n is the number of inputs. With two inputs there are $2^2 = 4$ possible input conditions. If we treat the inputs as bits in a multibit binary word, we can quickly define and record all input states by using the binary number equivalents. For example, with four possible input states we use the four numbers 0 through 3 or 00, 01, 10, and 11 in binary. These are all possible combinations for a two bit word.

A logic circuit with three inputs has $2^3 = 8$ different input conditions. We represent the eight possible input conditions with the binary equivalent of the numbers 0 through 7. If the inputs are designated X, Y, and Z, the input states are as shown below.

INPUTS			OUTPUT
X	Y	Z	W
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

This particular method of listing all possible input combinations by counting from zero through the upper limit is orderly and convenient.

There are two basic ways of using truth tables in logic work. First we can develop a truth table from a given logic circuit, and second we can develop an equation or logic circuit from a given truth table. The circuit-to-table method is useful in circuit analysis. The table-to-equation or circuit is a useful design method.

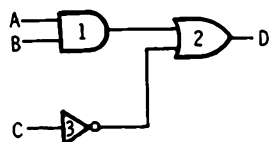


Figure 5-27

Developing the truth table for a given circuit starts with defining all of the inputs. Then the output for each gate in the circuit is developed until the final output is obtained. Consider the circuit shown in Figure 5-27. The Boolean output expression for this circuit is $D = AB + \bar{C}$. The output of gate 1 is AB and the output of inverter 3 is \bar{C} . These signals are ORed in gate 2 to produce $D = AB + \bar{C}$.

To develop the truth table for this circuit, we start with the inputs. Since there are three inputs there are $2^3 = 8$ possible input conditions that we define with the three bit numbers 000 through 111 as indicated in the table below. In the table we also create a column for the output of each gate or element in the circuit.

INPUTS			OUTPUTS		
A	B	C	Gate 1 AB	Inverter 3 \bar{C}	Gate 2 D
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

Observing the inputs, we can fill in the AB and \bar{C} columns. The \bar{C} is simply the complement of the C input column. To complete the AB column, you consider the A and B inputs and their effect on the AND gate output. Next, to find the D output, you note that the AB and \bar{C} outputs are ORed in gate 2. You complete the D column by recording a binary 1 each time either one or both of the AB or \bar{C} columns are binary 1. Check this in the table above.

Now, consider another example. Let's create a complete truth table for the circuit in Figure 5-28. The results are shown below.

INPUTS			OUTPUTS		
D	E	F	(D + E)	(E + F)	G
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

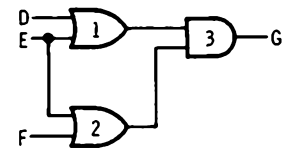


Figure 5-28

To produce this table you first note the number of inputs. There are three different variables D, E, and F so there are $2^3 = 8$ possible input combinations.

Next you create a column for each gate output. In this case, gate 1 output is (D + E), gate 2 output is (E + F), and the final output at gate 3 is $G = (D + E)(E + F)$.

To fill in the (D + E) column you OR together the D and E input columns. You record a binary 1 when either one or both column D or E are binary 1. You complete the (E + F) column in a similar manner considering all eight E and F input combinations.

Finally, you AND together the (D + E) and (E + F) columns, recording a binary 1 in the G column only when both (D + E) AND (E + F) columns are binary 1.

The truth table like the one you developed above completely defines the circuit operation for all possible conditions. And your analysis is thorough because to obtain the final output you had to derive the outputs of all other gates in the circuit.

In designing digital circuits we use the opposite approach. We develop a truth table as the result of our design. In designing a logic circuit we designate the number of inputs and what the output state should be for each set of input conditions. Then we write the Boolean equation from the truth table. From there the equation is readily translated into a logic diagram.

Assume that the circuit we want to design has two inputs A and B and we want a binary 1 output C to occur when A is 0 and B is 1 or when A is 1 and B is 0. Otherwise the output is binary 0.

The truth table outlining these conditions is shown below.

INPUTS		OUTPUT
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

To write the Boolean equation from this table we observe column C, noting those input states where binary 1s occur in the output. Then we write a sum-of-product Boolean expression based on these inputs. For each binary 1 state in the output we write one product term for the equation then we logically sum (OR) these terms.

The product terms are written from the input states. If the input is binary 1 we write the input letter. If the input is binary 0 we write the complement of the input letter. For example, when A is 0 and B is 1, the product term is $\bar{A}B$. Where A is 1 and B is 0, the product term is $A\bar{B}$. The input conditions for each binary 1 output state are now defined. The product terms are then ORed. The result is $C = \bar{A}B + A\bar{B}$.

Take another example. Write the output equation from the truth table below.

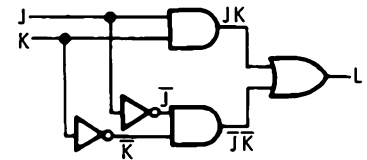
INPUTS		OUTPUT
J	K	L
0	0	1
0	1	0
1	0	0
1	1	1

The correct equation is $L = \bar{J}\bar{K} + JK$. A binary 1 output occurs for two of the four possible input combinations. Therefore you know that the sum-of-product output equation will include two product terms. The inputs are ANDed to form these products. The first output occurs when $J = 0$ and $K = 0$. Therefore the corresponding product is $\bar{J}\bar{K}$. The second output is binary 1 when $J = 1$ and $K = 1$. The product defining this condition is JK . To obtain the complete output expression we logically sum (OR) these products. $L = \bar{J}\bar{K} + JK$.

From here the logic diagram can be drawn. See Figure 5-29.

Now let's try a more complex problem. Write the equation and draw the circuit corresponding to the truth table below.

INPUTS			OUTPUT
A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



$$L = \bar{J}\bar{K} + JK$$

Figure 5-29

In this problem there are three inputs, so each product term will have three factors and the number of product terms is set by the number of times a binary 1 output appears in the D output column. You develop each product by looking at the input states and writing the variable when a binary 1 input occurs and the complemented variable when a binary 0 input occurs. The product term when $A = 0$, $B = 0$, and $C = 1$ is $\bar{A}\bar{B}C$. Once all the product terms are developed they are ORed together to obtain the output. The final equation is $D = \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$.

The logic diagram is then drawn from the equation. See Figure 5-30. Note in the logic diagram that inputs A, B, and C are considered to be available and signals \bar{A} , \bar{B} , and \bar{C} are generated with inverters. In some circuits the complement signals may already be available from other sources in which case the inverters can be omitted.

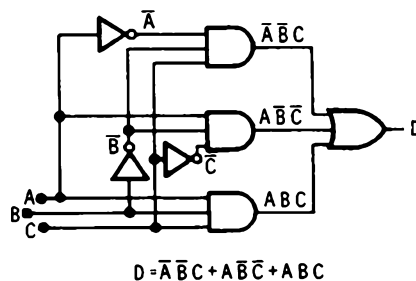


Figure 5-30

Self Test Review

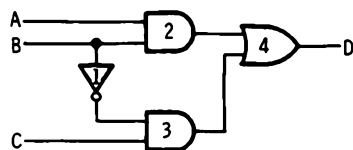


Figure 5-31

6. Develop a truth table for the circuit shown in Figure 5-31.
7. What is the Boolean equation corresponding to the 1 outputs in the truth table below?

INPUTS			OUTPUT
R	S	T	V
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Answers

6.

INPUTS			OUTPUTS			
A	B	C	Inverter 1 \bar{B}	gate 2 AB	gate 3 $\bar{B}C$	gate 4 D
0	0	0	1	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

7. $V = \bar{R}\bar{S}\bar{T} + \bar{R}ST + R\bar{S}\bar{T} + RST$

BOOLEAN RULES

As we mentioned earlier, the primary benefit of Boolean algebra to a technician or engineer today is in analyzing, understanding, and concisely expressing digital logic functions. The availability of a wide variety of integrated circuits has greatly minimized the use of Boolean algebra as a design tool. However, even with modern ICs, the designer can often benefit from the use of Boolean algebra in minimizing or implementing a function.

Boolean algebra is the algebra of two-valued functions. Many of the ordinary rules of algebra such as factoring or expanding a function, apply to Boolean expressions. However, the binary nature of the functions greatly simplifies most of the operations. There are also numerous special rules that apply to handling binary logic functions. We will explain these rules in this section and show how they are used.

The Laws of Intersection, for example, apply to AND gates. The two forms of this law are stated below.

$$A \cdot (1) = A$$

$$A \cdot (0) = 0$$

Remembering that A is a binary signal that can be either binary 0 or binary 1, we can prove the validity of these expressions if we remember how an AND gate works. The first expression simply says that if we apply a binary 1 to one input of an AND gate and the signal A to the other input, the output will be A. The binary 1 input simply enables the gate so that the A input state controls the output. If $A = 1$, the output will be 1. If $A = 0$, the output will be 0.

The circuit in Figure 5-32 expresses this relationship.



Figure 5-32

The other form of the Laws of Intersection is just as easy to understand.

$$A(0) = 0$$

It says that if one input to an AND gate is 0 and the other is A, the output will always be 0. Remember that the only time the output of an AND gate can be 1 is when all inputs are binary 1. If one input is fixed at 0, the output will always be 0. The circuit in Figure 5-33 expresses this. The Law of Intersection works for AND gates with more than two inputs. For example, $D \cdot E \cdot (1) = D \cdot E$ and $D \cdot E \cdot (0) = 0$.



Figure 5-33

Another similar set of rules exist for OR gates. These are called the Laws of Union. Expressed algebraically they are:

$$\begin{aligned} B + 1 &= 1 \\ B + 0 &= B \end{aligned}$$

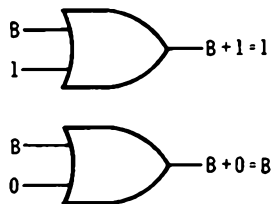


Figure 5-34

These rules are illustrated in Figure 5-34. These expressions, like those for the AND gate, almost perfectly define the operation of an OR gate. In the first version, if we apply a binary 1 to one input of an OR gate and a signal B to the other, the output will always be binary 1 by definition of an OR gate. If one input to an OR gate is binary 0, the other input B will control the output.

The Laws of Tautology apply to both AND gates and OR gates. The basic rules are given below.

$$\begin{aligned} A \cdot A &= A \\ B + B &= B \end{aligned}$$

The logic symbols illustrate these rules are shown in Figure 5-35.

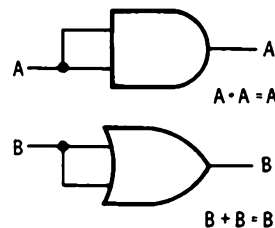


Figure 5-35

What these expressions say is that if you apply the same signal to all inputs of a logic gate, the output will be the same as the input. You can prove this to yourself by looking at the truth tables for AND and OR gates.

We can use the Laws of Tautology to simplify the expression $QT = JMX + JMX + F9$. Since $JMX + JMX = JMX$, $QT = JMX + F9$. The JMX term is redundant. In terms of circuitry you can see the simplification. The circuit in Figure 5-36A implements the original expression. Figure 5-36B shows the simplified but fully equivalent circuit.

This same simplification procedure applies to AND gates.

Consider the expression $X = Q \cdot Q \cdot J$. The original and simplified logic diagrams implementing this equation are shown in Figure 5-37. The original circuit can be implemented as shown in Figure 5-37A. The simplified but equivalent circuit is shown in Figure 5-37B. The logic function is identical. Now you are beginning to see that the value of Boolean algebra is simplifying the design of a circuit.

Another Boolean law is the Law of Complements. The two versions are

$$A \cdot \bar{A} = 0$$

$$B + \bar{B} = 1$$

If we apply a logic signal and its complement to a logic gate, the output becomes either a binary 0 or a binary 1 depending on type of logic gate. This is illustrated in Figure 5-38.

If you look closely at these circuits and analyze what happens, you will see that in either case the output is not affected by the state of the input. It can be either a 1 or a 0 and the output will be binary 0 for the AND gate and binary 1 for the OR gate. You can further verify these laws by referring to the AND and OR truth tables. A binary 0 on either input of an AND gate will always produce a binary 0 output. A binary 1 on either input of an OR gate will always produce a binary 1 output.

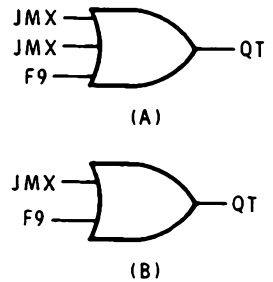


Figure 5-36

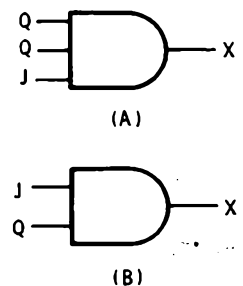


Figure 5-37

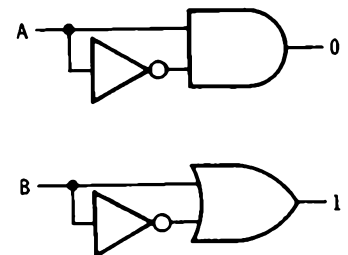


Figure 5-38

Now consider the Law of the Double Negative.

$$\overline{\overline{A}} = A$$

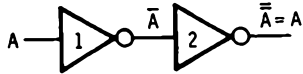


Figure 5-39

It says that the complement of the complement of A is equal to A . Or a signal that is complemented twice is the same as the original signal. You can see this from the circuit in Figure 5-39. If input A is 1, the output of inverter 1 is 0 and the output of inverter 2 is 1. Complementing a signal twice, or any even number of times, gives us the original signal.

For simplicity, if you encounter two inverters in cascade, you can simply remove them and substitute a piece of wire since the effect is the same. Three cascade inverters or any odd number produces the same effect as a single inverter.

Another Boolean rule is the Law of Commutation. This is the same rule from basic algebra. The two forms of it are:

$$\begin{aligned} A \cdot B &= B \cdot A \\ A + B &= B + A \end{aligned}$$

All it really says is that you can arrange the inputs to AND or OR gates in any order and the effect is the same. You can write the input variables in any order and they will mean the same thing.

$$W + X + Y = X + W + Y = Y + W + X$$

Also $JML = LJM = MLJ = JLM = LMJ = MJL$. The order of the inputs are different, but the logic result is the same.

Now let's give you some practice in using these rules. What are the simplifications of the expressions below?

- $A + \overline{B} + A = \underline{\hspace{2cm}}$
- $B C \overline{B} = \underline{\hspace{2cm}}$
- $C + 1 + \overline{B} = \underline{\hspace{2cm}}$
- $\overline{X} + Y + X = \underline{\hspace{2cm}}$

Work these problems using the previously discussed rules. The correct answers are given in Figure 5-40.

- a. $A + \bar{B}$
- b. 0
- c. 1
- d. 1

In problem (a) you first rearranged it by using the Law of Commutation $A + \bar{B} + A = A + A + \bar{B}$. By the Law of Tautology, you know that $A + A = A$, so the resulting simplified equation is $A + \bar{B}$.

In problem (b) you again rearrange the term using the Law of Commutation. $BC\bar{B} = B\bar{B}C$. From the Law of Complements you know that $B\bar{B} = 0$. Substituting this in the expression gives $0 \cdot C$. You know from the Law of Intersection that $0 \cdot C$ or $C \cdot 0 = 0$.

In problem (c) you should recognize the Laws of Union. $C + 1 + \bar{B} = 1$.

In problem (d), you first rearrange the equation by way of the Law of Commutation to $\bar{X} + Y + X = X + \bar{X} + Y$. The Law of Complements says that $X + \bar{X} = 1$. Therefore, the equation becomes $1 + Y$. Of course this reduces to 1 by way of the Law of Union.

Figure 5-40

Now try a few more examples using the previous solutions as a guide. But this time draw the logic diagram of the original expression as well as the simplified version to get a feel for the result of Boolean simplification in terms of circuitry.

- a. $L + M + \bar{M}$
- b. $JK\bar{K}L$
- c. $F + T + F$
- d. $\bar{B} + A C \bar{A} + D$

The answers are given in Figure 5-41.

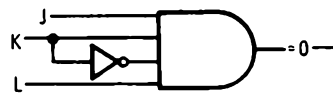
a. $L + M + \bar{M} = L + 1 = 1$

The three input OR and inverter called for by the original expression reduces to a single wire with a binary 1 on it.

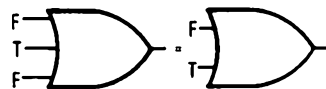


b. $JK\bar{K}L = J \cdot 0 \cdot L = 0$

The reduced expression is simply a binary 0 level.



c. $F + T + F = F + F + T = F + T$



d. $\bar{B} + AC\bar{A} + D = \bar{B} + A\bar{A}C + D = \bar{B} + 0 \cdot C + D = \bar{B} + 0 + D = \bar{B} + D$

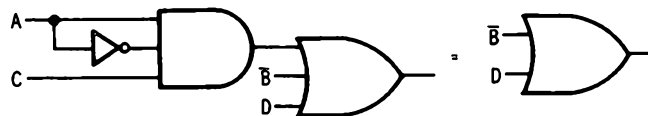


Figure 5-41

By now you should be able to see the value of Boolean algebra in simplifying a logic expression and minimizing the amount of circuitry required to implement it. Fewer parts means lower cost, smaller size and less power consumption.

Another Boolean law identical to the basic algebra rule is the Law of Association. These are:

$$(A \cdot B)C = A(B \cdot C) = A \cdot B \cdot C$$

$$A + (B + C) = (A + B) + C = A + B + C$$

You can see the circuitry simplification that results by applying these rules. See Figure 5-42. Note that we trade two 2-input AND gates for a single 3-input gate. The OR gate circuit equivalents for the Law of Association are given in Figure 5-43.

Note that the logical effect is the same for either version of the circuit. The three input gate is usually more economical than two 2 input gates.

Now let's consider the Laws of Distribution. We will use some of the rules described earlier to prove this law. The Laws of Distribution are:

$$A B + A C = A (B + C)$$

You should be able to infer this because all you are doing is factoring out one term (A) as you would in basic algebra.

The logic diagrams for each side of the above equation are shown in Figure 5-44.



Figure 5-42

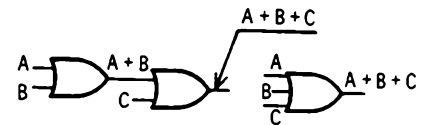


Figure 5-43

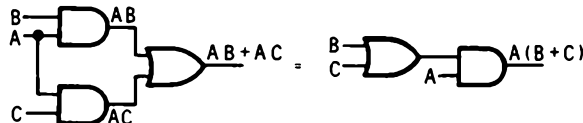


Figure 5-44

Not only is there a reduction in the number of circuits used, but also note that the expression changes from a sum-of-products to a product-of-sums form.

You can also use a truth table to show that the logic effect of both circuits is identical.

The table appears as shown below.

INPUTS			GATE OUTPUTS					
A	B	C	AB	AC	AB + AC	A	B + C	A (B + C)
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0
1	0	0	0	0	0	1	0	0
1	0	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1

The table shows the eight possible input combinations of the three variables. It also shows the outputs of the intermediate and output gates for both the sum-of-products and product-of-sums forms. And they are equal. This shows you another use of truth tables in analyzing and understanding logic circuits.

Another version of the Laws of Distribution is

$$(A + B)(A + C) = A + BC$$

Note the format of the original and simplified versions.

The original expression is in the product-of-sums form but is changed to the sum-of-products form when is is simplified. The logic diagrams of the two versions illustrated in Figure 5-45 show this difference.

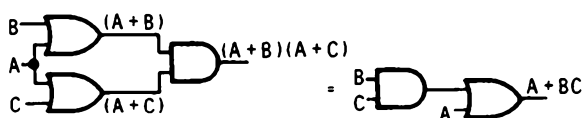
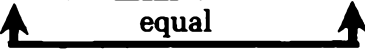


Figure 5-45

You can use truth tables to prove the equality of the two expressions or circuits.

See the table below.

INPUTS			GATE OUTPUTS					
A	B	C	(A + B)	(A + C)	(A + B) (A + C)	A	BC	A + BC
0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	0	1
1	0	1	1	1	1	1	0	1
1	1	0	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1



Another way to prove the equality of these expressions is to use some of the Boolean rules learned earlier.

$$(A + B)(A + C) = A + BC$$

First expand the expression on the left side of the equation as you would any algebraic expression. Multiply each term by the others and sum the products as shown.

$$(A + B)(A + C) = AA + AC + AB + BC$$

Next, this can be reduced by substituting A for AA and factoring an A out of the first three terms.

$$A + AC + AB + BC = A(1 + C + B) + BC$$

Then, the Laws of Union will reduce $(1 + C + B)$ to 1 and the expression becomes

$$A(1) + BC$$

Finally, the Laws of Intersection makes this

$$A + BC$$

Next there are the Laws of Absorption. There are four versions.

$$\begin{aligned} A(A + B) &= A \\ A(\bar{A} + B) &= AB \\ AB + B &= B \\ A\bar{B} + B &= A + B \end{aligned}$$

Now let's use the previously explained Boolean rules to prove the first expression.

$$A(A + B) = A$$

$$A(A + B)$$

Expand by multiplying (Distribution)

$$AA + AB$$

Replace AA by A (Tautology)

$$A + AB$$

Factor out A

$$A(1 + B)$$

Replace (1 + B) by 1 (Union)

$$A(1)$$

Replace A(1) by A (Intersection)

$$A = A(A + B)$$

Next let's prove $A(\bar{A} + B) = AB$

$$A(\bar{A} + B)$$

Expand by multiplying (Distribution)

$$A\bar{A} + AB$$

Replace $A\bar{A}$ by 0 (Complements)

$$0 + AB$$

Replace $AB + 0$ with AB (Union)

$$AB = A(\bar{A} + B)$$

The other Laws of Absorption are more difficult to prove. You can use truth tables as we did before. But it can be done with Boolean Algebra by using a trick. Let's prove that

$$AB + \bar{B} = A + \bar{B}$$

There isn't anything we can do with the expression on the right side of the equation. So let's multiply the \bar{B} term by $(A + 1)$. Since $A + 1 = 1$ and $\bar{B}(1) = \bar{B}$ we will not change the original meaning of the expression. It is equivalent to multiplying a term by 1. Therefore

$$AB + \bar{B} = AB + \bar{B}(A + 1)$$

Now we will use Boolean rules to further reduce this expression.

$$AB + \bar{B}(A + 1)$$

Expand by multiplying (Distribution)

$$AB + A\bar{B} + \bar{B}$$

Factor A out of first two terms (Distribution)

$$A(B + \bar{B}) + \bar{B}$$

Replace $(B + \bar{B})$ with 1 (Complements)

$$A(1) + \bar{B}$$

Replace $A(1)$ by A (Intersection)

$$A + \bar{B}$$

You can use the same trick to prove the expression.

$$A\bar{B} + B = A + B$$

$$A\bar{B} + B$$

Multiply B by $(A + 1)$ (Distribution)

$$A\bar{B} + B(A + 1)$$

Expand by multiplying (Distribution)

$$A\bar{B} + AB + B$$

Factor A out of first two terms (Distribution)

$$A(\bar{B} + B) + B$$

Replace $(\bar{B} + B)$ by 1 (Complements)

$$A(1) + B$$

Replace $A(1)$ by A (Intersection)

$$A + B = A\bar{B} + B$$

DeMORGAN'S THEOREM

Another important Boolean rule is DeMorgan's theorem. There are two basic forms of it as indicated below.

$$\overline{AB} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A}\overline{B}$$

One way to prove the equality of these expressions is by a truth table. The first expression above is proven this way as indicated below.

INPUTS		OUTPUTS				
A	B	\overline{A}	\overline{B}	AB	\overline{AB}	$\overline{A} + \overline{B}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

↑ equal ↑

There are two variables A and B so there are four possible input combinations. These are indicated in columns A and B. Columns are also provided for the other terms called for by the expressions

$$\overline{A}, \overline{B}, \overline{AB} \text{ and } (\overline{A} + \overline{B})$$

Using the inputs as a guide these other columns are completed. Go through each column yourself to be sure you understand how each state is obtained. Note the equality of the \overline{AB} and $\overline{A} + \overline{B}$ columns.

Now let's prove the expression $\overline{A + B} = \overline{A}\overline{B}$.

INPUTS		OUTPUTS				
A	B	\overline{A}	\overline{B}	A + B	$\overline{A + B}$	$\overline{A}\overline{B}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

↑ equal ↑

Remember if A = 0, $\overline{A} = 1$.

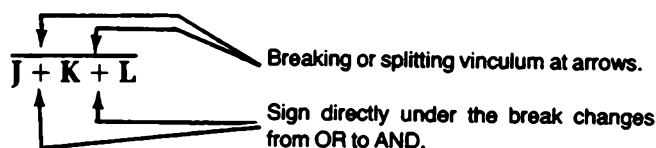
The correct proof of the expression $\overline{A + B} = \overline{A}\overline{B}$ is shown above.

Like other Boolean rules, DeMorgan's theorem is useful in minimizing logic equations. For example, the expression

$$X = \overline{A\overline{B}C} + \overline{A + \overline{C}}$$

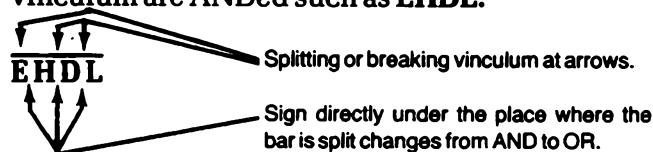
cannot be simplified by first using the previously given Boolean rules. It can only be reduced by using DeMorgan's theorem first, then the previously given Boolean rules.

DeMorgan's theorem is used to split or join vincula (complement bar). When a vinculum is split, every sign over which the splitting takes place changes from OR to AND or AND to OR and the bars are distributed over the individual terms. Let's consider an example where factors are ORed.



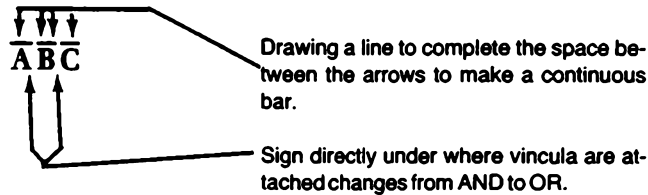
The result is, $\overline{J}\overline{K}\overline{L}$.

Let's consider an example where the factors in an expression under the vinculum are ANDed such as $\overline{EHD\overline{L}}$.



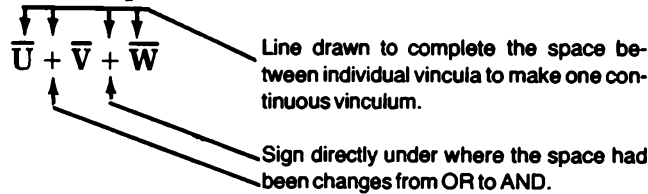
The result is, $\overline{E} + \overline{H} + \overline{D} + \overline{L}$

When vincula over individual factors or groups of factors are joined, the signs change from AND to OR or OR to AND. Consider an example where the factors are ANDed.



The result is: $\overline{A + B + C}$

An example where the factors are ORed is as follows:



The result is: $\overline{U V W}$

Another rule to remember when using DeMorgan's theorem is; when you change signs in an expression, group the same letters that were grouped originally. For example,

$\overline{A B + C}$ becomes $(\overline{A B})\overline{C} = (\overline{A} + \overline{B})\overline{C} = \overline{A}\overline{C} + \overline{B}\overline{C}$

Notice in the above example that the AND sign between factor A and factor B is a sign of grouping. Breaking the vinculum results in a sign change from AND to OR. Since the resulting OR sign is not a sign of grouping, parenthesis or brackets are used to indicate that the factors A and B were previously grouped.

Make sure you always follow the "signs of grouping" rules; otherwise, you will have errors. If we take the same example $\overline{A B + C}$ and do not apply the rule (do not place brackets or parenthesis around AB before breaking the vinculum), we would get the following:

Break Vinculum	$\overline{A B' + C}$	This result indicates that B is grouped with C instead of A.
Becomes	$\overline{A} + \overline{B} \overline{C}$	

When the rule is properly applied, we have

	$\overline{A B + C}$	
Insert grouping	$\overline{(A B) + C}$	
Break Vinculum	$\overline{(A B)' + C}$	A and B are still grouped as before
becomes	$(\overline{A} + \overline{B})\overline{C}$	
Multiplying	$\overline{A}\overline{C} + \overline{B}\overline{C}$	

Another type of expression dealing with signs of grouping is, $\overline{(Y + Z)X + W}$. In this example, X is grouped with the entire expression $Y + Z$ within the parenthesis. In many cases such as this, you could use any of the previously presented Boolean rules to simplify the expression before splitting the vinculum. Your procedure should be as follows:

• Temporarily ignore vinculum $\overline{(Y + Z)X + W}$

• In this case, apply the law of distribution to clear parenthesis. $\overline{XY + XZ + W}$

• Split vinculum at OR signs and change sign of operation. $\overline{XY} \overline{XZ} \overline{W}$

Note: There are two signs of grouping in the expression up to this point. The vinculum over XY and XZ as well as the AND signs between X and Y and X and Z.

Insert appropriate grouping signs $[\overline{XY}] [\overline{XZ}] \overline{W}$

• Split vinculum $[\overline{X} + \overline{Y}] [\overline{X} + \overline{Z}] \overline{W}$

• Law of Distribution $[\overline{X}\overline{X} + \overline{X}\overline{Z} + \overline{X}\overline{Y} + \overline{Y}\overline{Z}] \overline{W}$

• Idempotent (Tautology) $[\overline{X} + \overline{X}\overline{Z} + \overline{X}\overline{Y} + \overline{Y}\overline{Z}] \overline{W}$

• Factor out \overline{X} $[\overline{X}(1 + \overline{Z} + \overline{Y}) + \overline{Y}\overline{Z}] \overline{W}$

• Law of Union $[\overline{X}(1) + \overline{Y}\overline{Z}] \overline{W}$

• Law of Absorption $[\overline{X} + \overline{Y}\overline{Z}] \overline{W}$

• Law of Distribution $\overline{W}\overline{X} + \overline{W}\overline{Y}\overline{Z}$

The two major facts to remember about the above procedure is:

1. Watch your signs of grouping.
2. Simplify expression under vinculum as much as possible before splitting vinculum.

There is an alternate method which often can result in fewer steps. Let's use the same example $\overline{(Y + Z)X + W}$. However, instead of applying the law of distribution to remove the parenthesis as we did previously, we will add a sign of grouping.

add brackets: $\overline{[(Y + Z)X] + W}$

Split vinculum, but be careful to retain signs of grouping.

$$[(\bar{Y}\bar{Z}) + \bar{X}]\bar{W}$$

Law of Distribution: $\bar{W}\bar{Y}\bar{Z} + \bar{W}\bar{X}$

This second example gives the same result in fewer steps. However, given another expression, the first technique may require fewer steps. The main skill to achieve is accuracy. Only with a lot of practice can you hope to acquire the knack of knowing which procedure will give you an accurate result with the least amount of steps.

Let's look at the result when the same expression, $(Y + Z)X + W$ is simplified by ignoring signs of grouping and splitting vinculum immediately without retaining groupings by adding brackets.

$$\begin{array}{c} \downarrow \quad \downarrow \quad \downarrow \\ \overline{(Y + Z)X + W} \\ \bar{Y}\bar{Z} + \bar{X}\bar{W} \end{array}$$

Split vinculum above OR and AND signs

Notice that by splitting the vinculum in the initial step, the factors X and W are grouped together as if the original expression was $(Y + Z)(X + W)$. The result, although arrived at in one step, is obviously wrong because $(Y + Z)X + W$ is **not** equal to $(Y + Z)(X + W)$.

Before going on to some additional rules for the proper use of DeMorgan's theorem, practice using the first three DeMorgan's rules previously covered.

Self Test Review

Apply DeMorgan's theorem to the following expressions.

8. $\overline{DEFG} =$ _____
9. $\overline{E + F + G} =$ _____
10. $\overline{M} + \overline{N} + \overline{P} + \overline{R} =$ _____
11. $\overline{H} \overline{J} \overline{K} =$ _____
12. $\overline{D + EF} =$ _____
13. $\overline{R} + \overline{S} + \overline{T} + \overline{U} =$ _____
14. $\overline{(H + K) L} =$ _____
15. $\overline{R + LM} =$ _____
16. $\overline{A + BCD} =$ _____
17. $\overline{W} \overline{X} (\overline{Y} + \overline{Z}) =$ _____
18. $\overline{A + B(C + D)} =$ _____
19. The signs or methods used to show grouping in Boolean expressions are:

Answers

8. $\overline{DEFG} = \overline{D} + \overline{E} + \overline{F} + \overline{G}$
9. $\overline{E + F + G} = \overline{E} \overline{F} \overline{G}$
10. $\overline{M + N + P + R} = \overline{MNPR}$
11. $\overline{HJK} = \overline{H + J + K}$
12. $\overline{D + EF} = \overline{D + (EF)} = \overline{D} (\overline{E} + \overline{F}) = \overline{D} \overline{E} + \overline{D} \overline{F}$
13. $\overline{R + S + T + U} = \overline{RSTU}$
14. $(\overline{H} + \overline{K}) \overline{L} = \overline{HK + L}$
15. $\overline{R + LM} = \overline{R + (LM)} = \overline{R} (\overline{L} + \overline{M}) = \overline{R} \overline{L} + \overline{R} \overline{M}$
16. $\overline{A + BCD} = \overline{A + (BCD)} = \overline{A} (\overline{B} + \overline{C} + \overline{D}) = \overline{A} \overline{B} + \overline{A} \overline{C} + \overline{A} \overline{D}$
17. $\overline{WX} (\overline{Y} + \overline{Z}) = (\overline{W} \overline{X}) (\overline{Y} + \overline{Z}) = \overline{(W + X) + (YZ)}$

Note in number 10 the rules for signs of grouping apply when joining vincula as well as splitting a vinculum.

18. $\overline{A + B(C + D)} = \overline{A} \overline{B} + \overline{A} \overline{C} \overline{D}$
 First, add sign of grouping: $\overline{A + [B(C + D)]}$

$$\begin{array}{c} \downarrow \quad \downarrow \quad \downarrow \\ \overline{A + [B(C + D)]} \end{array}$$

 Split the vinculum at places shown:
 Be careful to retain signs of grouping: $\overline{A} [\overline{B} + (\overline{C} \overline{D})]$
 Now use the law of distribution: $\overline{A} \overline{B} + \overline{A} \overline{C} \overline{D}$

19. The signs or methods used to show grouping in Boolean expressions are:

AND	AB	OR	A + B
Brackets	[AB]		[A + B]
Parenthesis	(AB)		(A + B)
Vinculum	\overline{AB}		$\overline{A + B}$

DeMORGAN'S THEOREM (Continued)

If a vinculum covers part of an expression such as $\overline{B(A + CDE)}$ the following simplification rules apply for splitting vinculum:

- Signs outside the vinculum **do not** change.
- Signs under the vinculum **do** change.

Consider the following example:

$$\overline{B(A + CDE)}$$

Place signs of grouping,

$$\overline{B[A + (CDE)]}$$

Split vinculum where indicated,

$$\begin{array}{c} \downarrow \quad \downarrow \downarrow \\ \overline{B[A + (CDE)]} \\ \overline{B}[\overline{A}(\overline{C} + \overline{D} + \overline{E})] \end{array}$$

At this point, you should recall a rule in ordinary algebra that also applies to Boolean algebra. Where there are signs of grouping within signs of grouping, clear the inside grouping first. In this case, apply the law of distribution to clear inner parenthesis,

$$\overline{B}[\overline{A}\overline{C} + \overline{A}\overline{D} + \overline{A}\overline{E}]$$

Law of Distribution again,

$$(\overline{B}\overline{A}\overline{C} + \overline{B}\overline{A}\overline{D} + \overline{B}\overline{A}\overline{E})$$

If a vinculum covers part of an expression such as $B + \overline{CD}$, the following simplification rules, when **joining** vinculum, apply:

- You can not extend a vinculum over a factor or term that does not have a vincula or vinculum. For example, the expression,

$$B + \overline{C(D + E)}$$

cannot be changed to

$$\overline{B[C(D + E)]}.$$

$\overline{B[C(D + E)]}$ is equal to $\overline{B} + \overline{C(D + E)}$ but is not equal to the original expression $B + \overline{C(D + E)}$.

- To extend a vinculum over a factor or factors that do not have a vinculum, the law of double negative ($A = \overline{\overline{A}}$) must first be applied.

Let's take the same example used above, $B + \overline{C(D + E)}$

Apply Double Negative $\overline{\overline{B}} + \overline{C(D + E)}$

Join the **upper** vincula to original vinculum $\overline{\overline{B}[C(D + E)]}$
and change sign of operation.

$$\overline{\overline{B}[C(D + E)]} = \overline{\overline{B}} + \overline{C(D + E)} = B + \overline{C(D + E)}$$

Let's consider some other ways DeMorgan's theorem can be used with additional laws to simplify expressions. Always try to manipulate an expression so that one part is the complement of the other. For example,

$$\begin{array}{l} \text{Apply DeMorgan's to } \overline{A} + \overline{B} \end{array} \quad \begin{array}{l} AB + \overline{A} + \overline{B} + CD \\ AB + \overline{AB} + CD \end{array}$$

Now you have AB and its complement \overline{AB} (which is equivalent to $A + \overline{A} = 1$) so the expression can be simplified to $1 + CD$.

$$\text{Law of Union, } 1 + CD = 1$$

$$\text{Suppose the expression was} \quad A + B + CD + \overline{AB}$$

$$\text{Law of Commutation} \quad A + B + \overline{AB} + CD$$

$$\text{DeMorgan's} \quad A + B + \overline{A} + \overline{B} + CD$$

$$\text{Law of Commutation} \quad A + \overline{A} + B + \overline{B} + CD$$

$$\text{Law of Complements} \quad 1 + 1 + CD$$

$$\text{Law of Union} \quad 1 + 1 + CD = 1$$

Consider the following expression where the double negative law can be used to advantage.

$$\overline{(L + K)(J + M)}$$

DeMorgan's
Double Negative
DeMorgan's

$$\begin{aligned} &\overline{\overline{L + K} + \overline{J + M}} \\ &\overline{L + K} + J + M \\ &\overline{L} \overline{K} + J + M \end{aligned}$$

An alternate method in simplifying the same expression, but using the double negative law twice, is as follows:

$$\overline{(L + K)(J + M)}$$

Place double negative
over $L + K$

$$\overline{\overline{(L + K)}(J + M)}$$

Join upper vinculum over
 $L + K$ to vinculum over $J + M$,
and change sign of operation
Double Negative
DeMorgan's

$$\begin{aligned} &\overline{\overline{(L + K)} + (J + M)} \\ &\overline{L + K} + J + M \\ &\overline{L} \overline{K} + J + M \end{aligned}$$

Remember, unless you can manipulate an expression so that you have a group of letters and its complement (so that both groups taken together equal 0 or 1), the best procedure for simplifying is:

- First, split or join the vincula according to DeMorgan's theorem.
- Group letters as they were originally grouped.
- Apply other laws, in whatever order seems best.

To give you the necessary practice in using Boolean Laws and DeMorgan's theorem, take the following self-test review. The answers are given after the problems. However, try to work these problems before referring to the answers. Some of the expressions can be reduced more than one way.

Self Test Review

Simplify the following expressions.

20. $\overline{A}B\overline{C} + \overline{A}\overline{B}C$
21. $(A + \overline{B} + \overline{C})(A + \overline{C})$
22. $\overline{E}E + CD + \overline{B} + A + E(\overline{CD} + A)$
23. $D + [E + \overline{F}(G + H)][\overline{E}F(G + H)]$
24. $[W + \overline{X}Y + \overline{W}(X + \overline{Y})]Z$
25. $\overline{(AB + C)}(\overline{A} + \overline{B})\overline{C}$
26. $\overline{X}\overline{Y} + X\overline{Z}$

Self Test Answers

20. 1
 DeMorgan's $\overline{ABC + \overline{A}BC}$
 Law of Complements $\overline{ABC \cdot \overline{A}BC}$
 Law of Intersection $\overline{0 \cdot BC \cdot BC}$ $[A \cdot \overline{A} = 0]$
 $\overline{0}$ $[0 \cdot A = 0]$
 Complement of 0 = 1

21. \overline{ABC}
 DeMorgan's $\overline{(A + \overline{B} + \overline{C})(A + \overline{C})}$
 Law of Association $\overline{(A + \overline{B} + \overline{C}) + (A + \overline{C})}$
 Law of Commutation $\overline{A + \overline{B} + \overline{C} + A + \overline{C}}$
 Idempotent Law $\overline{A + A + \overline{C} + \overline{C} + \overline{B}}$
 $\overline{A + \overline{B} + \overline{C}}$ $[A \cdot A = A]$
 De Morgan's \overline{ABC}

22. 1
 There are two approaches to minimizing this expression. Either one should result in the same answer. The first method is the standard approach beginning with DeMorgan's theorem and requires many steps:

$$\overline{E}E + CD + \overline{B} + A + E(\overline{CD} + A)$$

DeMorgan's.

Watch signs of grouping

- | | |
|---------------------|---|
| | $\overline{E} + CD + \overline{B} + A + E[(\overline{C} + \overline{D})\overline{A}]$ |
| Law of Distribution | $\overline{E} + CD + \overline{B} + A + E[\overline{A}\overline{C} + \overline{A}\overline{D}]$ |
| Law of Distribution | $\overline{E} + CD + \overline{B} + A + \overline{A}\overline{C}E + \overline{A}\overline{D}E$ |
| Law of Absorption | $\overline{E} + CD + \overline{B} + A + \overline{A}\overline{C} + \overline{A}\overline{D}$ |

Law of Absorption	$\overline{E} + CD + \overline{B} + A + \overline{C} + \overline{D}$	
Law of Commutation	$\overline{E} + \overline{C} + CD + A + \overline{D} + \overline{B}$	
Law of Absorption	$\overline{E} + \overline{C} + (D + \overline{D}) + A + \overline{B}$	
Law of Complements	$\overline{E} + \overline{C} + 1 + \overline{B} + A$	$[D + \overline{D} = 1]$
Laws of Union	$1 + \overline{C} + \overline{B} + \overline{E} + A = 1$	$[1 + B = 1]$

The second method is much quicker because it has fewer steps.

Law of Commutation	$\overline{E}\overline{E} + CD + \overline{B} + A + E(\overline{CD} + A)$
Law of Double Negative	$\overline{E} + CD + A + E(\overline{CD} + A) + \overline{B}$
DeMorgan's	$\overline{E} + CD + A + \overline{\overline{E}(\overline{CD} + A)} + \overline{B}$
Law of Complements	$\overline{E} + CD + A + \overline{\overline{E} + CD + A} + \overline{B}$ $\quad \quad \quad \underbrace{\quad \quad \quad}_{A + \overline{A} = 1} \quad \quad \quad = 1 + \overline{B}$
Law of Union	$1 + \overline{B} = 1$

23. D	$D + [E + \overline{F(G+H)}][\overline{EF(G+H)}]$
Law of Double Negative	$D + [\overline{\overline{E} + \overline{F(G+H)}}][\overline{EF(G+H)}]$
DeMorgan's	$D + \underbrace{[\overline{\overline{E} + \overline{F(G+H)}}]}_{\overline{A}} \cdot \underbrace{[\overline{EF(G+H)}]}_{A} = 0$
Law of Complements	$D + \overline{A} \cdot A = 0$
Law of Intersection	$D + 0 = D$

24. Z	$[W + \overline{X}Y + \overline{W}(X + \overline{Y})]Z$
Double Negative	$[W + \overline{X}Y + \overline{W}(\overline{\overline{X}} + \overline{\overline{Y}})]Z$
DeMorgan's	$[\overline{W + \overline{X}Y} + \overline{W + \overline{X}Y}]Z$
Law of Complements	$[A + \underbrace{\overline{A}}_{=1}]Z$
Law of Intersection	$1 \cdot Z = Z$

25. $AB + C$	$\overline{(\overline{AB + C})(\overline{A + B})C}$
DeMorgan's	$\overline{(\overline{AB + C})(\overline{A + B})}C$
Idempotent (Tautology)	$\underbrace{\overline{(\overline{AB + C})(\overline{A + B})}}_{\overline{A} \cdot \overline{A} = A}C = A$
Result Therefore is:	$\overline{\overline{AB + C}}$
Double Negative	$(\overline{\overline{A}} = A) = AB + C$

26. $XZ + \bar{X}Y$

DeMorgan's

$$\overline{\bar{X}Y + XZ}$$

DeMorgan's

$$(\overline{\bar{X}Y})(\overline{XZ})$$

Double Negative

$$(\bar{X} + \bar{Y})(\bar{X} + \bar{Z})$$

Law of Distribution

$$(X + Y)(\bar{X} + Z)$$

Law of Complements

$$X\bar{X} + XZ + \bar{X}Y + YZ$$

Multiply YZ term by 1

$$XZ + \bar{X}Y + YZ$$

Law of Distribution

$$XZ + \bar{X}Y + YZ(X + \bar{X})$$

Law of Commutation

$$XZ + \bar{X}Y + XYZ + \bar{X}YZ$$

Law of Absorption

$$XZ + XYZ + \bar{X}Y + \bar{X}YZ$$

Factor XZ and $\bar{X}Y$

$$XZ(1 + Y) + \bar{X}Y(1 + Z)$$

Law of Intersection

$$XZ(1) + \bar{X}Y(1)$$

$$XZ + \bar{X}Y$$

MINIMIZING LOGIC CIRCUITS

You have already seen how the Boolean rules are used to simplify logic expressions and you have had a little practice in minimizing some logic equations yourself. But now we want you to polish your skill with Boolean so that you can minimize any logic circuit that you might encounter. The discussion presented here will give you the necessary practice.

Let's start with the expression $F = (\bar{A} + B)(B + \bar{C})$. The logic diagram for this expression is given in Figure 5-46.

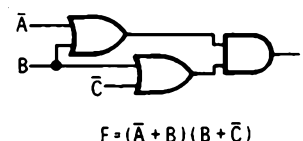


Figure 5-46

Now, using Boolean algebra, let's minimize the equation. The first step is to expand the expression by multiplying as you would in algebra.

$$F = (\bar{A} + B)(B + \bar{C}) = \bar{A}B + \bar{A}\bar{C} + BB + B\bar{C}$$

Basic Boolean rules can then be used as indicated below to reduce the equation.

$$\bar{A}B + \bar{A}\bar{C} + BB + B\bar{C}$$

Laws of Tautology (Idempotent) $(B \cdot B) = B$

$$\bar{A}B + \bar{A}\bar{C} + B + B\bar{C}$$

Laws of Commutation (rearrange equation)

$$\bar{A}B + B + B\bar{C} + \bar{A}\bar{C}$$

Laws of Absorption
(factor out B) $B(\bar{A} + 1 + \bar{C}) + \bar{A}\bar{C}$
 $B(1) + \bar{A}\bar{C}$

$$B + \bar{A}\bar{C}$$

The logic diagram of the minimized version of the original expression is shown in Figure 5-47.

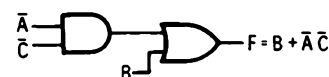


Figure 5-47

As you can see, there is a significant simplification in that the minimized circuit uses one less gate.

Just to be sure that the two circuits and expressions do indeed produce the same logical function, we can prove their equality with a truth table. The truth table proves conclusively that

$$F = (\bar{A} + B)(B + \bar{C}) = B + \bar{A}\bar{C}$$

The original and simplified expressions are equivalent.

INPUTS			GATE OUTPUTS						
A	B	C	\bar{A}	\bar{C}	$(\bar{A} + B)$	$(B + \bar{C})$	$(\bar{A} + B)(B + \bar{C})$	$\bar{A}\bar{C}$	$B + \bar{A}\bar{C}$
0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	0	0	0	0
0	1	0	1	1	1	1	1	1	1
0	1	1	1	0	1	1	1	0	1
1	0	0	0	1	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0
1	1	0	0	1	1	1	1	0	1
1	1	1	0	0	1	1	1	0	1

↑ equal ↑

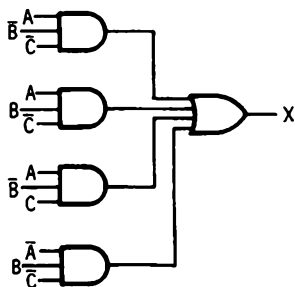


Figure 5-48

Next let's try the expression below.

$$X = A\bar{B}\bar{C} + AB\bar{C} + A\bar{B}C + \bar{A}B\bar{C}$$

The logic diagram is shown in Figure 5-48.

Observing the equation, try to spot common factors in each of the terms, then regroup the terms and rearrange the factors using the Laws of Commutation.

$$\begin{aligned} X &= A\bar{B}\bar{C} + AB\bar{C} + A\bar{B}C + \bar{A}B\bar{C} \\ X &= A\bar{B}\bar{C} + A\bar{B}C + B\bar{C}A + B\bar{C}\bar{A} \end{aligned}$$

Next we factor out the common expressions, $A\bar{B}$ in the first two terms and $B\bar{C}$ in the last two terms.

$$\begin{aligned} X &= A\bar{B}(\bar{C} + C) + B\bar{C}(A + \bar{A}) \\ &= A\bar{B}(1) + B\bar{C}(1) \end{aligned}$$

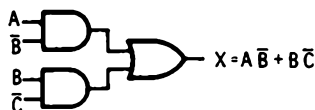


Figure 5-49

By the Laws of Complements this becomes:

$$X = A\bar{B} + B\bar{C}$$

The corresponding logic diagram is given in Figure 5-49.

Comparing this circuit with the one in Figure 5-48, you can see the result of minimization. The three gate circuit performs exactly the same logic function as the original five gate circuit at a considerable savings.

Next, let's minimize a logic equation with DeMorgan's theorem.

Reduce the equation

$$G = \overline{\overline{\overline{A} \overline{B} \overline{C}} + BC} (\overline{A} \overline{B})$$

The correct solution is shown below.

$$G = \overline{\overline{\overline{A} \overline{B} \overline{C}} + BC} (\overline{A} \overline{B})$$

DeMorgan's

$$G = \overline{\overline{A} \overline{B} \overline{C}} + BC + \overline{A} \overline{B}$$

DeMorgan's

$$G = \overline{A} + B + C + BC + \overline{A} + B$$

Law of Commutation

$$G = \overline{A} + \overline{A} + B + B + C + BC$$

Law of Tautology

$$G = \overline{A} + B + C + BC$$

Law of Distribution

$$G = \overline{A} + B + C(1 + B)$$

Law of Union

$$G = \overline{A} + B + C(1)$$

Law of Intersection

$$G = \overline{A} + B + C$$

Self Test Review

Simplify the following logic equations using Boolean Algebra.

$$27. \quad X = AB\bar{C}D + ABC\bar{D} + B\bar{C}D + \bar{A}BC\bar{D}$$

$$28. \quad F = (A + B + \bar{C})(\bar{A} + B + \bar{C})$$

$$29. \quad X = \frac{A\bar{B}C + A + \bar{C}}{\quad}$$

$$30. \quad M = (A + \bar{B})(\bar{A} + C)(B + C)$$

$$31. \quad D = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C} + ABC$$

Answers

$$27. \quad X = AB\bar{C}D + ABC\bar{D} + B\bar{C}D + \bar{A}BC\bar{D}$$

Rearrange order of terms (Law of Commutation)

$$X = AB\bar{C}D + B\bar{C}D + ABC\bar{D} + \bar{A}BC\bar{D}$$

Factor out $B\bar{C}D$ in first two terms and $BC\bar{D}$ in second two terms (Law of Distribution)

$$X = B\bar{C}D(A + 1) + BC\bar{D}(A + \bar{A})$$

Reduce first term by Law of Union and the second by Law of Complements

$$X = B\bar{C}D(1) + BC\bar{D}(1)$$

Reduce by Law of Intersection

$$X = B\bar{C}D + BC\bar{D}$$

Factor out B (Law of Distribution)

$$X = B(\bar{C}D + C\bar{D})$$

$$28. \quad F = (A + B + \bar{C})(\bar{A} + B + \bar{C})$$

Expand by multiplying (Law of Distribution)

$$A\bar{A} + AB + A\bar{C} + B\bar{A} + BB + B\bar{C} + \bar{C}\bar{A} + \bar{C}B + \bar{C}\bar{C}$$

Minimize with Laws of Tautology and Complements.

$$F = AB + A\bar{C} + B\bar{A} + B + B\bar{C} + \bar{C}\bar{A} + \bar{C}B + \bar{C}$$

Rearrange terms (Law of Commutation)

$$F = A\bar{C} + B\bar{C} + \bar{A}\bar{C} + \bar{C} + AB + \bar{A}B + B$$

Factor out terms (Law of Distribution)

$$F = \bar{C}(A + B + \bar{A} + 1) + B(A + \bar{A} + 1)$$

Reduce by Law of Union

$$F = \overline{C}(1) + B(1)$$

Reduce by Law of Intersection and rearrange terms (Law of Commutation)

$$F = B + \overline{C}$$

$$29. \quad X = \overline{A\overline{B}C} + \overline{A + \overline{C}}$$

Expand with DeMorgan's theorem - Double negative rule

$$X = \overline{A} + \overline{\overline{B}} + \overline{\overline{C}} + \overline{A\overline{C}} = \overline{A} + B + \overline{C} + \overline{A}C$$

Rearrange terms (Law of Commutation)

$$X = \overline{A} + \overline{A}C + B + \overline{C}$$

Factor out \overline{A} (Law of Distribution)

$$X = \overline{A}(1 + C) + B + \overline{C}$$

Reduce by Law of Union

$$X = \overline{A}(1) + B + \overline{C}$$

Reduce by Law of Intersection

$$X = \overline{A} + B + \overline{C}$$

$$30. \quad M = \overline{(A + \overline{B})(\overline{A} + C)(B + C)}$$

Expand by Law of Distribution starting with first two terms.
Ignore vinculum for now.

$$M = \overline{(A\overline{A} + AC + \overline{A}\overline{B} + \overline{B}C)(B + C)}$$

Law of Complements $A\overline{A} = 0$

$$M = \overline{(AC + \overline{A}\overline{B} + \overline{B}C)(B + C)}$$

Expanded by Law of Distribution

$$M = \overline{ABC + ACC + \overline{A}\overline{B}B + \overline{A}\overline{B}C + \overline{B}CB + \overline{B}CC}$$

Law of Complements $\overline{A}\overline{B}B = 0, \overline{B}CB = 0$

$$M = \overline{ABC + ACC + \overline{A}\overline{B}C + \overline{B}CC}$$

Law of Tautology (idempotent)

$$M = \overline{ABC + AC + \overline{A}\overline{B}C + \overline{B}C}$$

Law of Absorption Factor out AC and $\overline{B}C$

$$M = \overline{AC + \overline{B}C}$$

DeMorgan's Theorem

$$M = \overline{(\overline{AC})(\overline{\overline{B}C})}$$

DeMorgan's and Double negative

$$M = \overline{(\overline{A} + \overline{C})(B + \overline{C})}$$

Law of Distribution

$$M = \overline{A}B + \overline{A}\overline{C} + B\overline{C} + \overline{C}\overline{C}$$

Law of Tautology (idempotent)

$$M = \overline{A}B + \overline{A}\overline{C} + B\overline{C} + \overline{C}$$

Law of Absorption, Factor \overline{C}

$$M = \overline{A}B + \overline{C}(\overline{A} + B + 1)$$

Law of Union ($A + 1 = 1$)

$$M = \overline{A}B + \overline{C}(1)$$

Law of Intersection

$$M = \overline{A}B + \overline{C}$$

31. $D = \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C} + ABC$

Law of Commutation

$$D = \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC + ABC$$

Factor out $B\overline{C}$ and BC

$$D = B\overline{C}(\overline{A} + A) + BC(\overline{A} + A)$$

Law of Complements

$$D = B\overline{C}(1) + BC(1)$$

Law of Intersection

$$D = B\overline{C} + BC$$

Factor out B

$$D = B(\overline{C} + C)$$

Law of Complements

$$D = B(1)$$

Law of Intersection

$$D = B$$

Compare the equation $D = \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C} + ABC$ to the truth table. A and C have no effect on the output, and thus, the original logic circuit reduces to a single switch marked B .

TRUTH TABLE			
INPUTS			OUTPUTS
A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	$\overline{A}B\overline{C}$
0	1	1	$\overline{A}BC$
1	0	0	0
1	0	1	0
1	1	0	$AB\overline{C}$
1	1	1	ABC

USING NAND/NOR GATES

Throughout this unit, we have shown logic equations implemented with AND and OR gates. However, most modern digital systems and circuits are made with NAND or NOR gates. As you saw in an earlier unit, any of the three basic logic operations can be realized with either NAND or NOR circuits.

A NAND gate or a NOR gate can be used as an inverter by tying all of the inputs together as shown in Figure 5-50. Both circuits produce an output that is the complement of the input.

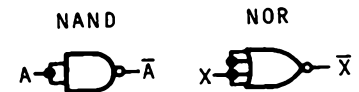


Figure 5-50

When used as inverters NAND and NOR gates can be represented by the inverter symbol. See Figure 5-51

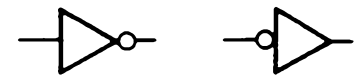


Figure 5-51

A NAND gate can be used for AND operations by inverting its output as shown in Figure 5-52. The inversion of the NAND is cancelled by the added inverter producing the pure AND function.



Figure 5-52

When you use inverters to complement the inputs to a NAND, the OR function will be performed. The output equation of this combination circuit, shown in Figure 5-53, is $C = \overline{\overline{A} \overline{B}}$. Using DeMorgan's theorem the output expression reduces to $A + B$.

DeMorgan's theorem tells that a NAND can also perform the negated OR function.

$$\overline{\overline{A} \overline{B}} = \overline{\overline{A}} + \overline{\overline{B}}$$

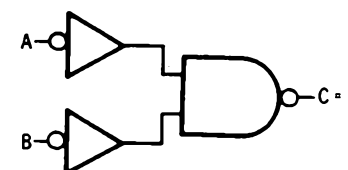


Figure 5-53

When the NAND is used for the negated OR function the symbol in Figure 5-54 is used. This makes it clear what function is being performed.



Figure 5-54

Putting inverters ahead of the inputs creates the pure OR function as indicated in Figure 5-55

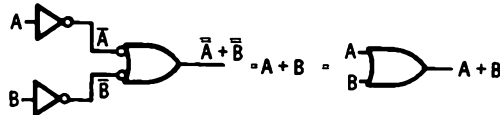


Figure 5-55

You should realize that the symbol used for the negated OR shows circles or inverters at the inputs. In a true physical sense there are no inverters there, but the logic effect produced is as if there were.

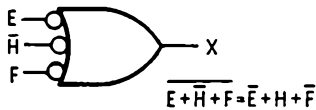


Figure 5-56

In writing the Boolean equation from a logic diagram simply treat the symbol literally and interpret each circle as an inversion and each gate as the designated logic function.

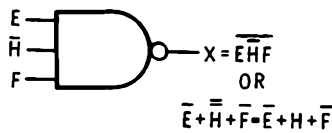


Figure 5-57

The equation of the circuit in Figure 5-56 is $\bar{E} + H + \bar{F}$. If you had drawn this circuit with the equivalent NAND symbol as indicated in Figure 5-57, you might expect a different logic operation to be performed. The equation for this circuit is $X = \overline{EHF}$.

Using DeMorgan's theorem on this expression it becomes $\bar{E} + H + \bar{F}$.

The NOR gate can also be used to perform any logic function. By inverting its output the OR operation is performed as shown in Figure 5-58.

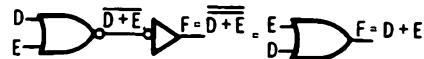


Figure 5-58

In an earlier unit you saw how using inverters at the input of a NOR could also be used to express the AND function. The logic symbols illustrating this are shown in Figure 5-59.

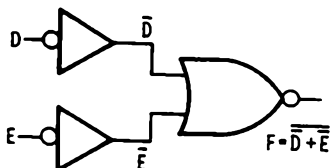


Figure 5-59

The output equation of this circuit is $F = \overline{\overline{D} + \overline{E}}$. Reducing it with DeMorgan's theorem we get $F = \overline{\overline{D} + \overline{E}} = \overline{\overline{D}} \cdot \overline{\overline{E}}$. This is of course the AND function $D \cdot E$.

DeMorgan's theorem tells us that the positive NOR can also perform as a negated AND. The DeMorgan's expression for this relationship is $\overline{D} + \overline{E} = \overline{D \cdot E}$.

When the negated AND function is used, the symbol in Figure 5-60 indicates the function.

The negated AND connected as a standard AND is shown in Figure 5-61.

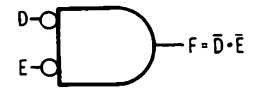


Figure 5-60

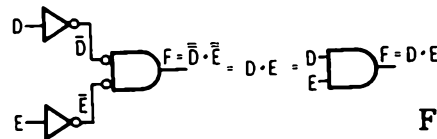
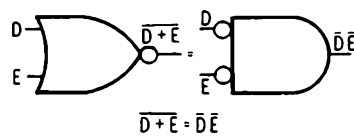
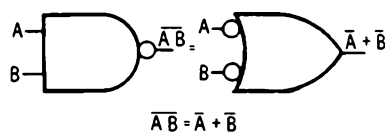


Figure 5-61

Interpret the negated AND logic symbol literally for the purpose of writing equations, even though in reality no inverters are physically present at the inputs where circles are indicated.

Figure 5-62 summarizes the functions and symbols of NAND and NOR gates.



NAND TRUTH TABLE

A	B	OUTPUT
L	L	H
H	L	H
L	H	H
H	H	L

D	E	OUTPUT
L	L	H
H	L	L
L	H	L
H	H	L

NOR TRUTH TABLE

Figure 5-62

The AND and OR equivalent circuits with NANDs and NORs are summarized in Figure 5-63

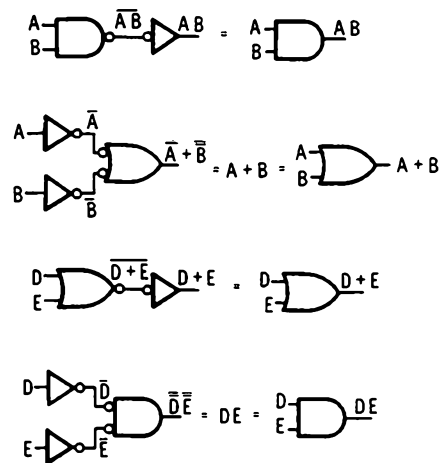


Figure 5-63

We will show how any Boolean equation can be implemented with NAND or NOR gates in the following section.

NOR LOGIC EQUIVALENT CIRCUITS

The procedure for implementing any Boolean expression with NOR gates is as follows:

Step 1. Minimize the equation using Boolean algebra.

Step 2. Draw the logic circuit of the reduced equation using standard AND and OR symbols.

The first example we will use is, $F = X + \bar{Y}Z$, which is already in its simplest form. The circuit is shown in Figure 5-64.

Step 3. Draw the output NOR gate showing the reduced equation at its output as in Figure 5-65.

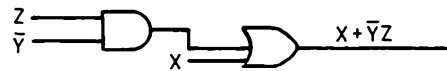


Figure 5-64

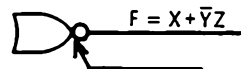


Figure 5-65

Note: With this method, we start at the output gate and the final output expression working from right to left (output to input). Both sections of the NOR gate must be considered but only one section at a time.

Step 4. Complement the output. Working from right to left in Figure 5-65, the inverter is the part of the NOR gate first encountered. Now you know that at the point indicated by the arrow in Figure 5-65, the signal must pass through the inverter to become $F = X + \bar{Y}Z$. Therefore, the expression describing the signal at the arrow is the complement or $X + \bar{Y}Z$. Figure 5-66 shows the results of this fourth step in our procedure.

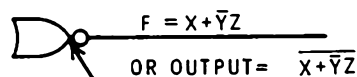


Figure 5-66

Step 5. Simplify the output expression of the OR gate. This is the expression shown at the arrow in Figure 5-66

$$F = \overline{X} + \overline{Y}Z$$

DeMorgan's $(\overline{X})(\overline{\overline{Y}Z})$

DeMorgan's $(\overline{X})(Y + \overline{Z})$

Distribution $\overline{X}Y + \overline{X}\overline{Z}$

Step 6. The result indicates two terms.

Therefore, the output gate is a 2-input NOR with $\overline{X}Y$ one input and $\overline{X}\overline{Z}$ the other input. The result is shown in Figure 5-67.

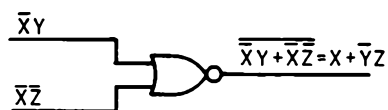


Figure 5-67

Step 7. Draw a NOR gate connected to each input line of the output NOR. This is shown in Figure 5-68.

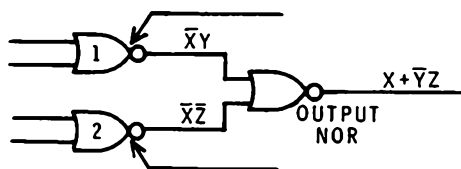


Figure 5-68

In steps 8 and 9 of this procedure, you must consider both sections of these NOR gates: The inverter, then the OR.

Step 8. Complement the output expressions of the NORs at the point indicated by the arrow in Figure 5-68, the signal must pass through the inverter to become $\overline{\overline{X}Y}$ for gate 1 and $\overline{\overline{X}\overline{Z}}$ for gate 2. The expression describing the signal at the arrow then is the complement. $\overline{\overline{X}Y}$ for gate 1. $\overline{\overline{X}\overline{Z}}$ for gate 2.

Step 9. Determine the NOR inputs by simplifying the complemented expression. To find the inputs for gate 1, you apply DeMorgan's theorem to $\overline{\overline{X}Y}$. The result is $X + \overline{Y}$. Applying DeMorgan's theorem to $\overline{\overline{X}Z}$ results in $X + \overline{Z}$, the inputs to NOR gate 2. The procedure in step 8 and 9 and the results are shown in Figure 5-69.

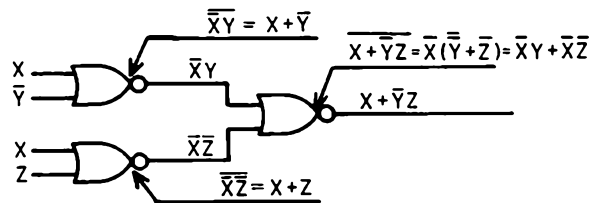


Figure 5-69

Step 10. Remove all but the input signals from your diagram.

Step 11. Using the inputs, determine the outputs of each gate. See Figure 5-70.

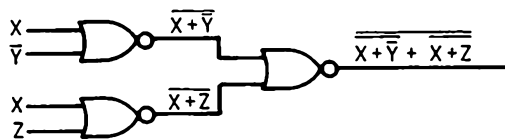


Figure 5-70

Step 12. Simplify the output shown in Figure 5-70 proving that the equivalent circuit is correct if the result is equal to the original equation, $X + \overline{Y}Z$.

Insert signs of grouping	$\overline{\overline{(X + \overline{Y})} + \overline{(X + Z)}}$
DeMorgan's	$\overline{(X + \overline{Y})} \overline{(X + Z)}$
Double Negative	$(X + \overline{Y}) (X + Z)$
Distributive	$XX + XZ + X\overline{Y} + \overline{Y}Z$
Idempotent (Tautology)	$X + XZ + X\overline{Y} + \overline{Y}Z$
Absorption	$X + \overline{Y}Z$

Let's try the example, shown in Figure 5-71.

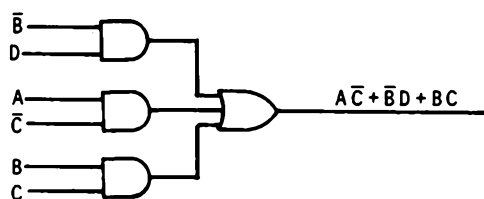


Figure 5-71

Since the circuit and expression are both already in simplified form, steps 1 and 2 of our procedure are complete, so we can begin with step 3, drawing the output NOR gate with its output equation. See Figure 5-72.

Step 3.

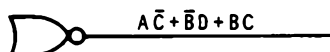


Figure 5-72

Step 4. Working from output to input (right to left), complement the output equation to show the action of the inverter. This is shown in Figure 5-73.

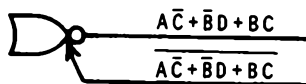


Figure 5-73

Step 5. Find inputs to the OR section by simplifying the complemented expression.

	$\overline{AC + BD + BC}$
	$\downarrow \downarrow \downarrow \downarrow \downarrow$
Place signs of grouping	$(\overline{AC}) + (\overline{BD}) + (\overline{BC})$
DeMorgan's	$(\overline{A} + C)(B + \overline{D})(\overline{B} + \overline{C})$
Distribution $B\overline{B} = 0$	$(\overline{A} + C)(B\overline{B} + B\overline{C} + \overline{B}\overline{D} + \overline{C}\overline{D})$
Complement	$(\overline{A} + C)(B\overline{C} + \overline{B}\overline{D} + \overline{C}\overline{D})$
Distribution	$\overline{A}B\overline{C} + \overline{A}B\overline{D} + \overline{A}\overline{C}\overline{D} + \overline{C}B\overline{C} + \overline{C}B\overline{D} + \overline{C}\overline{C}\overline{D}$
Complement and Idempotent	$\overline{A}B\overline{C} + \overline{A}B\overline{D} + \overline{A}\overline{C}\overline{D}(1) + \overline{B}C\overline{D}$
Intersection	$\overline{A}B\overline{C} + \overline{A}B\overline{D} + (\overline{A}\overline{C}\overline{D})(B + \overline{B}) + \overline{B}C\overline{D}$
Complements	$\overline{A}B\overline{C} + \overline{A}B\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{B}C\overline{D}$
Distribution Factor $\overline{A}B\overline{C}$ and $\overline{A}B\overline{D}$	$\overline{A}B\overline{C}(1 + \overline{D}) + \overline{A}B\overline{D}(1 + \overline{C}) + \overline{B}C\overline{D}$
Absorption	$\overline{A}B\overline{C} + \overline{A}B\overline{D} + \overline{B}C\overline{D}$

Step 6. Determine number of inputs and draw a NOR gate on each input. The result of step 5 above is an expression with three terms. This indicates that the output gate must have three inputs. The three inputs of the output gate each must be driven by a NOR gate. These conclusions are shown in Figure 5-74.

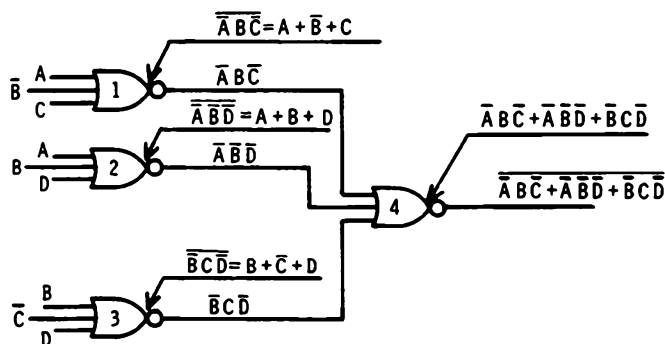


Figure 5-74

- Step 7. Now complement, each of the NOR gate outputs to account for the inverting part of each NOR. This is shown in Figure 5-74.
- Step 8. Using DeMorgan's theorem, simplify each of the complemented outputs as in Figure 5-74.
- Step 9. Complete the diagram in Figure 5-74 by showing the inputs to each of the NOR gates. The inputs are the results of step 8 above.
- Step 10. Draw the completed diagram showing only the input signals.
- Step 11. Now, using these inputs, determine the outputs of each gate to acquire the final output equation. See Figure 5-75.
- Step 12. This is a check step. Simplify the output equation. The simplified result should be the original output equation in Figure 5-71. If it is, the equivalent circuit in Figure 5-75 is accurate.
- Step 11.

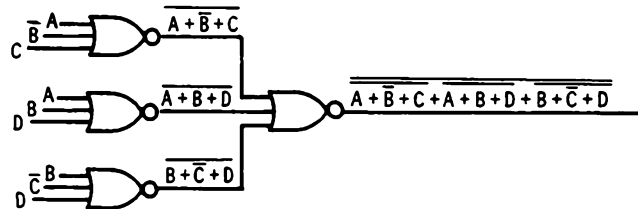


Figure 5-75

Step 12. STEP BY STEP SIMPLIFICATION

FROM FIGURE 5-75	$\overline{A + \overline{B} + C + A + B + D + B + \overline{C}D}$
DeMorgan's and Grouping Terms	$\overline{(A + \overline{B} + C)(A + B + D)(B + \overline{C}D)}$
Drop Double Negative	$(A + \overline{B} + C)(A + B + D)(B + \overline{C} + D)$
Expand Equation	$(AA + AB + AD + A\overline{B} + B\overline{B} + \overline{B}D + AC + BC + CD)(B + \overline{C} + D)$
Factor A	$[A(1 + B + D + \overline{B} + C) + (B\overline{B} + \overline{B}D + BC + CD)](B + \overline{C} + D)$
Union	$(A + B\overline{B} + \overline{B}D + BC + CD)(B + \overline{C} + D)$
Complement	$(A + \overline{B}D + BC + CD)(B + \overline{C} + D)$
Expand Equation	$AB + A\overline{C} + AD + \overline{B}BD + \overline{B}\overline{C}D + \overline{B}DD +$ $BBC + B\overline{C}C + BCD + BCD + C\overline{C}D + CDD$
Complements	$AB + A\overline{C} + AD + \overline{B}\overline{C}D + \overline{B}DD + BBC + BCD + BCD + CDD$
Idempotent	$AB + A\overline{C} + AD + \overline{B}\overline{C}D + \overline{B}D + BC + BCD + CD$
Factor $\overline{B}D$ and CD	$AB + A\overline{C} + AD + \overline{B}D(\overline{C} + 1) + CD(B + 1) + BC$
Union	$AB + A\overline{C} + AD + \overline{B}D + CD + BC$
Multiply by 1	$AB + A\overline{C} + AD + \overline{B}D + CD(B + \overline{B}) + BC$
Expand Equation	$AB + A\overline{C} + AD + \overline{B}D + BCD + \overline{B}CD + BC$
Factor $\overline{B}D$ and BC	$AB + A\overline{C} + AD + \overline{B}D(1 + C) + BC(D + 1)$
Union	$AB + A\overline{C} + AD + \overline{B}D + BC$
Multiply by 1	$AB + A\overline{C} + AD(B + \overline{B}) + \overline{B}D + BC$
Expand Equation	$AB + A\overline{C} + ABD + A\overline{B}D + \overline{B}D + BC$
Factor AB and $\overline{B}D$	$AB(1 + D) + A\overline{C} + \overline{B}D(A + 1) + BC$
Union	$AB + A\overline{C} + \overline{B}D + BC$
Multiply by 1	$AB(C + \overline{C}) + A\overline{C} + \overline{B}D + BC$
Expand Equation	$ABC + AB\overline{C} + A\overline{C} + \overline{B}D + BC$
Factor BC and $A\overline{C}$	$BC(A + 1) + A\overline{C}(B + 1) + \overline{B}D$
Union	$BC + A\overline{C} + \overline{B}D$ The original equation.

The key to multiplying by 1 is to ensure that each term contains the needed variables.

Single-Letter Term Expressions

One type of expression cannot be handled in the same way as those previously discussed. This type has all single-letter terms. For example, the expression, $E + D + H + \bar{F}$ contains all single-letter terms. It is neither a product-of-sums nor a sum-of-products expression. Let's determine the NOR circuit for this single-term expression.

1. Draw the output NOR as in Figure 5-76.

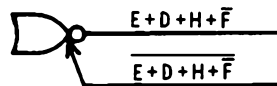


Figure 5-76

2. Complement the output expression.
3. Simplify the complemented expression to determine the NOR gate inputs using DeMorgan's theorem.

$$\overline{E + D + H + \bar{F}}$$

$$\bar{E}\bar{D}\bar{H}F$$

As you can see by the result of step 3, there is no way to split $\bar{E}\bar{D}\bar{H}F$ into two or more inputs. Therefore the correct way to diagram this type of expression is to perform steps 1 and 2 above as usual. Then, instead of simplifying the expression as in step 3 above, merely make the complemented expression a single input to the output NOR, which in turn, is connected as an inverter. See Figure 5-77.



Figure 5-77

Now, instead of using DeMorgan's theorem to simplify the complemented expression, as you previously have done, use the individual terms or factors under the vinculum as the inputs to a NOR. The completed circuit is shown in Figure 5-78.



Figure 5-78

Self Test Review

32. Write the equation for the circuit in Figure 5-79; simplify using Boolean algebra.
Draw the circuit for the simplified expression using AND and OR gates. Then, draw the NOR equivalent circuit.
33. Implement the logic expression $F = C(A + \bar{B})$ with conventional AND and OR gates. Then use all NOR gates.
34. Construct a NOR logic circuit for the expression, $\bar{A} + \bar{B} + C + \bar{D} + \bar{E}$.

Answers

32. $X = \overline{\overline{\overline{AB}}CD}$

DeMorgan's $X = \overline{\overline{\overline{AB}}(C)(\bar{D})}$

Law of Double Negative $X = (\overline{\overline{AB}})(C) + \bar{D}$

DeMorgan's $X = (\bar{A} + \bar{B})(C) + \bar{D}$

Distribution $X = AC + \bar{B}C + \bar{D}$

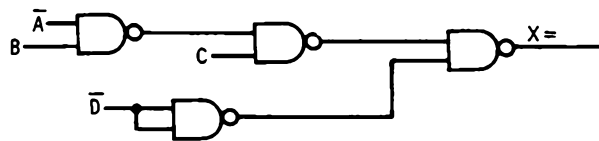


Figure 5-79

Note that the circuit of Figure 5-79 is drawn using all positive logic NAND symbols. The simplified circuit of Figure 5-80 has two versions.

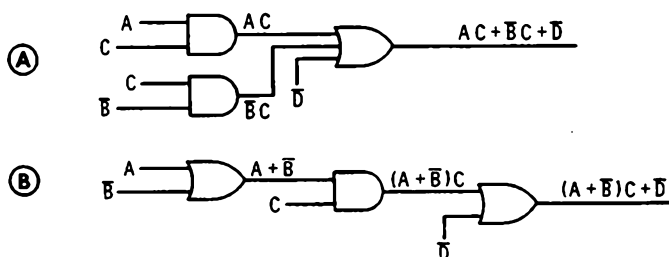


Figure 5-80

33. See Figure 5-81A and B.

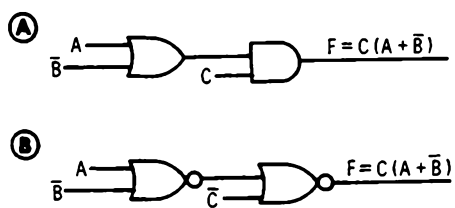


Figure 5-81

34. See Figure 5-82.

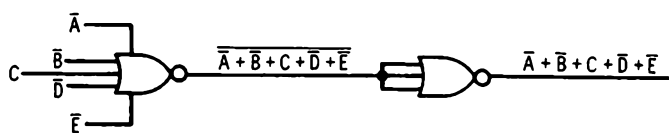


Figure 5-82

NAND LOGIC EQUIVALENT CIRCUITS

The procedure that is used in changing a logic circuit to an equivalent circuit using all NAND gates will depend on whether the output expression is in a product-of-sums or in a sum-of-products form. However, any one of the two procedures is much simpler than the previously discussed procedure implementing NOR gates. Let's begin with an example previously used when implementing NOR gates. It is the expression $X + \bar{Y}Z$ which is a sum-of-products form.

Sum of Products Equations

The expression $X + \bar{Y}Z$ is already in its simplest form and the circuit using standard AND and OR gates is shown in Figure 5-64. Therefore, the first two steps of the procedure are already accomplished, so we will proceed with the third step.

Step 3. Draw the output NAND gate showing the output expression. See Figure 5-83.

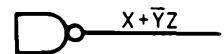


Figure 5-83

Step 4. Working from output to input, place a double vinculum over the output expression as shown in Figure 5-84.

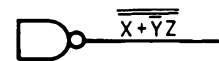


Figure 5-84

Step 5. Passing through the inverter symbol from right to left (from output to input), results in dropping one vinculum. The result is the output of the AND portion of the NAND gate. See Figure 5-85.

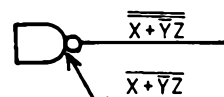


Figure 5-85

Now notice that the uppermost vinculum remains on the right side of the inverter because you are proceeding from right to left through one inverter. The left side of the inverter now has the complement of the original expression.

Step 6. Working with the expression at the output of the AND, break the vinculum just above the OR sign. Be careful with your signs of grouping. The result is $(\bar{X}) (\bar{Y} Z)$. The terms within the signs of grouping are the inputs to the output gate. See Figure 5-86.

There are two terms indicating that the output NAND will have two inputs. One single input another driven by a NAND.

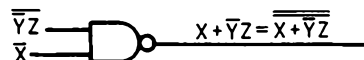


Figure 5-86

Step 7. Draw a NAND gate on the line that has the $\overline{\overline{Y}Z}$ expression. See Figure 5-87.

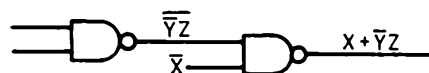


Figure 5-87

Step 8. Proceeding from output to input through the inverter of the NAND results in removing a vinculum. Therefore, the output of the AND portion of the NAND gate is $\overline{\overline{Y}Z}$. See Figure 5-88. Note $\overline{\overline{\overline{Y}Z}} = \overline{\overline{Y}Z}$.

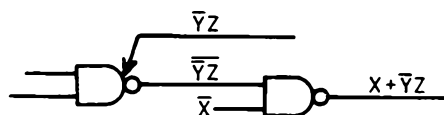


Figure 5-88

Step 9. Determine inputs. To get $\overline{\overline{Y}Z}$ at the output of the AND portion of the NAND, the inputs must be \overline{Y} and Z .

Step 10. Draw the equivalent NAND circuit showing inputs and all outputs as in Figure 5-89.

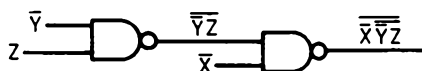


Figure 5-89

Step 11. Check Step. Simplify the output expression of Figure 5-89; the result should be the original expression.

$$\begin{aligned} \text{DeMorgan's} \quad & \overline{\overline{\overline{X} \overline{\overline{Y}Z}}} \\ & \overline{\overline{X}} + \overline{\overline{\overline{Y}Z}} \end{aligned}$$

$$\text{Double Negative} \quad X + \overline{\overline{Y}Z} = \text{Original expression.}$$

Let's try another expression previously used for NOR logic, $A\overline{C} + \overline{B}D + BC$. The circuit using standard AND and OR symbols is shown in Figure 5-71. Since the circuit and output expression are already in simplified form, we can start our conversion to all NAND gates by beginning at step 3.

Step 3. Draw the output NAND gate showing the simplified expression at its output. See Figure 5-90

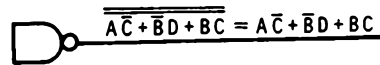


Figure 5-90

Step 4. Place a double vinculum over the output expression as shown in Figure 5-90.

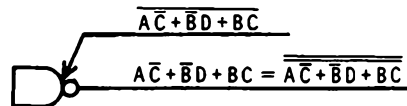


Figure 5-91

Step 5. Show the output of the AND portion of the NAND. Note if the output of the NAND is $A\bar{C} + \bar{B}D + BC$ then, after going through the inverter portion (proceeding from right to left), the expression must be $\overline{A\bar{C} + \bar{B}D + BC}$ which is the complement of the original expression. See the result in Figure 5-91.

Step 6. Use DeMorgan's theorem on the complemented expression by splitting the vinculum. Split the vinculum only at the OR signs as follows:

$$\text{DeMorgan's} \quad \overline{A\bar{C} + \bar{B}D + BC} = \overline{A\bar{C}} \cdot \overline{\bar{B}D} \cdot \overline{BC}$$

Step 7. Determine the number of inputs to the output NAND gate and draw a NAND for each. The result is three complemented terms. Each is the output of a NAND driving an input of the output NAND gate as shown in Figure 5-92.

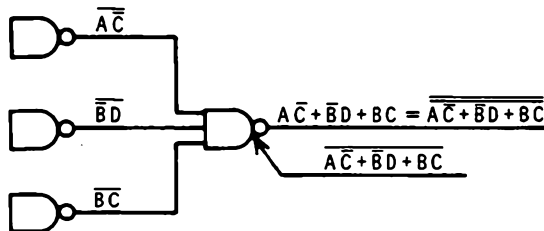


Figure 5-92

Step 8. Determine the output of the AND portion of each NAND gate.

Note: Proceeding from output to input of the inverters results in the complement of the $A\bar{C}$, $\bar{B}D$, and $\bar{B}\bar{C}$ terms at their respective AND gate outputs.

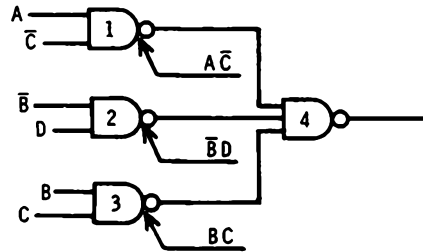


Figure 5-93

Step 9. Determine the inputs to each NAND gate. To get $A\bar{C}$, $\bar{B}D$ and $\bar{B}\bar{C}$ at the output of AND gates 1, 2 and 3, the inputs to these gates must be as shown in Figure 5-94.

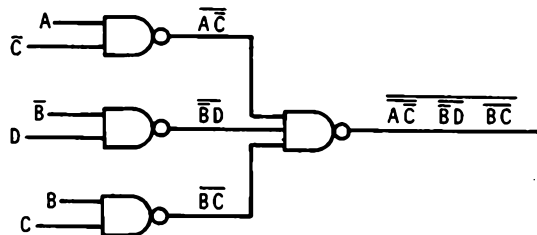


Figure 5-94

Step 10. Now, proceeding from inputs to outputs in the alleged completed diagram, show all input and output signals. When completed, your diagram should appear as in Figure 5-94.

Step 11. Prove that the diagram in Figure 5-94 is equivalent to Figure 5-71. Use DeMorgan's theorem and any other Boolean laws to simplify the output expression shown in Figure 5-94. The result will match the original expression if there were no mistakes in any of the previous operations.

	$\overline{\overline{A\bar{C}} \ \overline{\bar{B}D} \ \overline{\bar{B}\bar{C}}}$
DeMorgan's	$\overline{\overline{A\bar{C}} + \overline{\bar{B}D} + \overline{\bar{B}\bar{C}}}$
Double Negative	$A\bar{C} + \bar{B}D + \bar{B}\bar{C} = \text{Original expression.}$

Product-Of-Sums Expressions

The procedure for implementing NAND logic from product-of-sums expressions such as $(A + \bar{B})(C + \bar{D})$ is somewhat different than the procedure for sum-of-product equations. Let's use the expression $(A + \bar{B})(C + \bar{D})$. It is already in its simplest terms; therefore we will begin the procedure at the second step.

Step 2. Draw the circuit using standard AND and OR symbols. See Figure 5-95A.

Step 3. Draw the output NAND gate. See Figure 5-95B.

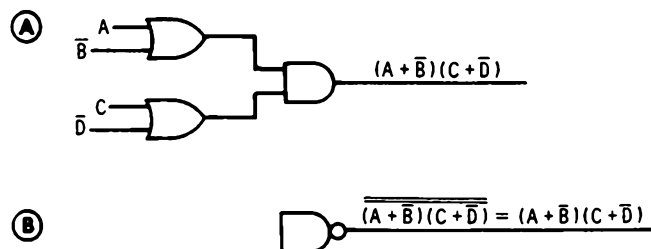


Figure 5-95

Step 4. Place a double vinculum over the output expression.

Step 5. Proceeding from output to input of the inverter show the output of the AND portion of the NAND gate.

Note: This will be the complement of the original expression. See Figure 5-96.

Step 6. Using DeMorgan's theorem simplify the expression at the AND gate output of Figure 5-96.

Note: This step is different than Step 6 for the sum-of-products expression. In that procedure the expression is partially simplified. In this case, you simplify completely to acquire the simplest form as follows:

$$\begin{array}{l} \text{DeMorgan's} \quad \overline{(A + \bar{B})(C + \bar{D})} \\ \text{DeMorgan's} \quad \overline{A + \bar{B} + C + \bar{D}} \\ \quad \quad \quad \bar{A}B + \bar{C}D \end{array}$$

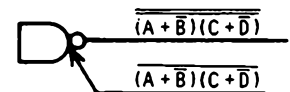


Figure 5-96

Step 7. Determine the inputs to the output NAND gate and draw a NAND gate on each of these inputs. See Figure 5-97.

Step 8. Show each term in the result of step 6 above as the output of the input NAND gates.

Step 9. Determine the inputs to the NAND gates. See Figure 5-97.

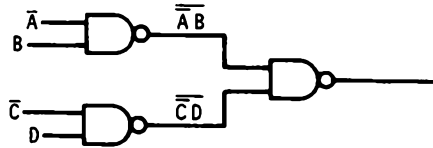


Figure 5-97

Step 10. Show inputs and outputs of all gates. See Figure 5-98.

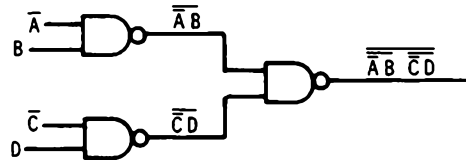


Figure 5-98

Step 11. This is a check step. As in the other procedures, the output expression is now simplified as a check step. If the simplification results in the original expression, the circuit is accurate and complete. So, let's perform the check.

DeMorgan's	$\overline{\overline{A} B + \overline{C} D}$
Double Negative	$\overline{A} B + \overline{C} D$

Notice that the result is not the original expression. So let's try again. However, instead of connecting the two vincula as we did previously, let's split the vincula.

	$\overline{\overline{A} B} \overline{\overline{C} D}$
DeMorgan's	$\overline{(\overline{A} + \overline{B}) (\overline{C} + \overline{D})}$
Signs of Grouping	$(\overline{\overline{A} + \overline{B}}) (\overline{\overline{C} + \overline{D}})$
Double Negative	$(A + B) (C + D)$

Now notice the expression matches the original except for the vinculum over it. Further simplification, DeMorgan's etc., will not result in the original expression. Try it. So what can be done to the circuit to eliminate the vinculum? Add an inverter as in Figure 5-99. Remember the double negative law says $\overline{\overline{(A + \overline{B}) (C + \overline{D})}} = (A + \overline{B}) (C + \overline{D})$

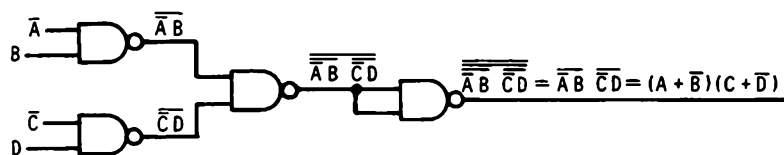


Figure 5-99

Once you become familiar with DeMorgan's theorem and its relationship with NAND and NOR gates, the procedure you have seen here will come naturally to you. A little practice and experience will make you competent in dealing with NAND and NOR integrated circuits to implement logic functions. Keep in mind that your goal is to arrive at the minimum circuit for your application.

Self Test Review

Given the diagram in Figure 5-100, do the following:

35. Determine the output expression.
36. Simplify the expression.
37. Diagram the simplified expression using standard AND and OR gates.
38. Diagram the simplified expression using all NAND gates.
39. Check your final result.

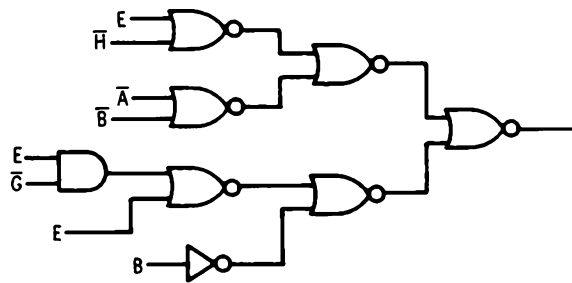


Figure 5-100

Answers

35. See Figure 5-101

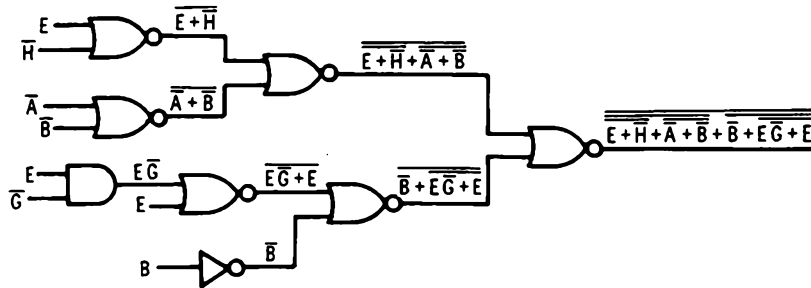


Figure 5-101

36. $AB\bar{E} + \bar{E}H$

1. Break vinculum where indicated and change sign of operation

$$\overline{\overline{E + H + A + B + B + EG + E}}$$

2. Drop double vinculum

$$[(E + H) + (\bar{A} + \bar{B})] [\bar{B} + (\bar{E}G + E)]$$

3. Break vinculum where indicated

$$[(\bar{E} + \bar{H}) + (\bar{A} + \bar{B})] [\bar{B} + (\bar{E}G + E)]$$

4. Drop double vinculum

$$[(\bar{E}\bar{H}) + (\bar{A}\bar{B})] [\bar{B} + (\bar{E} + \bar{G})(\bar{E})]$$

5. Expand the right side of the equation

$$[(\bar{E}H) + (AB)] [\bar{B} + (\bar{E} + G)\bar{E}]$$

6. Consolidate right side

$$[\bar{B} + (\bar{E}\bar{E} + \bar{E}G)]$$

7. Factor right side

$$[\bar{B} + (\bar{E} + \bar{E}G)]$$

8. G ORed with one equals one

$$[\bar{B} + (1 + G)\bar{E}]$$

9. Expand full equation

$$[\bar{E}H + AB] [\bar{B} + \bar{E}]$$

10. Consolidate $AB\bar{B} = 0$
 $\bar{E}\bar{E}H = \bar{E}H$

$$[\bar{B}\bar{E}H + AB\bar{B} + \bar{E}\bar{E}H + AB\bar{E}]$$

11. Factor $\bar{E}H$

$$[\bar{B}\bar{E}H + \bar{E}H + AB\bar{E}]$$

12. Consolidate $\bar{B} + 1 = 1$

$$\bar{E}H(\bar{B} + 1) + AB\bar{E}$$

13. Simplified equation

$$\bar{E}H + AB\bar{E}$$

37. See Figure 5-102.

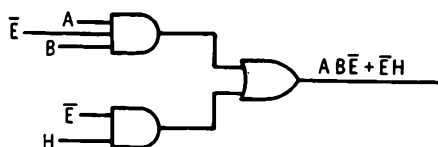


Figure 5-102

38. See Figure 5-103.

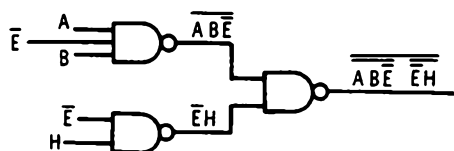


Figure 5-103

39. $AB\bar{E} + \bar{E}H$

$$\overline{AB\bar{E}} \quad \overline{\bar{E}H}$$

DeMorgan's

$$\overline{AB\bar{E} + \bar{E}H}$$

Double Negative

$$AB\bar{E} + \bar{E}H$$

EXPERIMENT 7

Applying NAND and NOR Gates

OBJECTIVES: *To show how TTL and CMOS, NAND, and NOR gates are used to implement any logic functions and to demonstrate the value of Boolean algebra in reducing logic circuits to their minimum configuration.*

Materials Required

Heathkit Digital Design Experimenter (ET-3200)

1 — 74LS00 TTL IC (443-728)

1 — 74LS20 TTL IC (443-798)

1 — 74LS02 TTL IC (443-779)

1 — 4001 CMOS IC (443-695)

Procedure

1. Write the output expression of the circuit shown in Figure 5-104.
F = _____

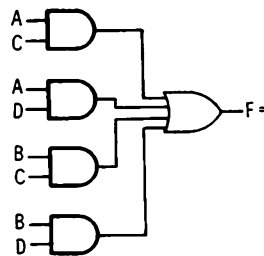


Figure 5-104

2. Figure 5-105 shows the NAND gate implementation of the circuit in Figure 5-104. Wire the circuit shown in Figure 5-104. The pin connections for the 74LS00 and 74LS20 IC are given in Figure 5-106. Be sure to connect pin 14 to +5 volts and pin 7 to GND on each IC. Connect +5V to all unused inputs.

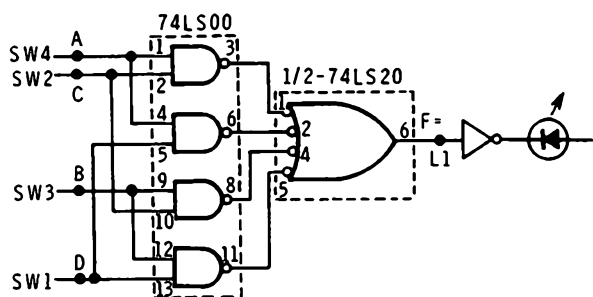


Figure 5-105

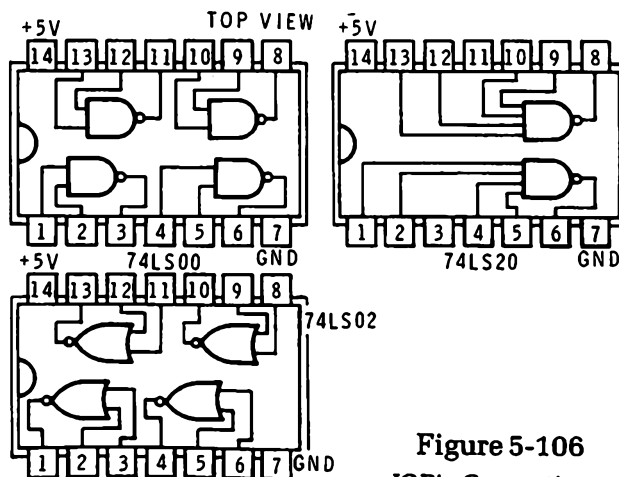


Figure 5-106
IC Pin Connections.

Note:

In figure 5-105 and 5-107, we show a dashed line around the IC gate symbols. This represents the IC package. The number (74LS00, 74LS20, 74LS02, etc.) identifies the type of IC. Accompanying the IC designation number you will see $\frac{1}{2}$, $\frac{3}{4}$ or other fractions. This is a method of indicating how many of the gates in the IC are used. For example, in a 74LS00 IC there are 4-2 input NANDs. In Figure 5-105 we use all four of them. The 74LS20 IC contains two - 4 input gates. But in Figure 5-105 we use only one of them and we indicate this by the designation $\frac{1}{2}$ -74LS20. In Figure 5-107 we use three of the four NORs in the 74LS02 so the designation is $\frac{3}{4}$ -74LS02. This terminology will be used throughout the program.

3. Apply the inputs A, B, C and D in Table I to the circuit with data switches SW1 through SW4. Monitor the output on L1 and record the state for each set of inputs in the left-hand F column in Table I.

INPUTS				OUTPUTS	
A (SW4)	B (SW3)	C (SW2)	D (SW1)	F (L1) Fig. 5-105	F (L1) Fig. 5-107
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Table I

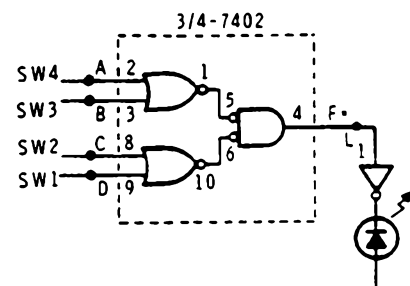


Figure 5-107

4. Using Boolean algebra, reduce the output equation obtained in step 1. The minimized expression is:

$$F = \underline{\hspace{2cm}}$$

5. Construct the circuit shown in Figure 5-107. Remember to connect +5V to all unused inputs.
6. Write the output equation of the circuit in Figure 5-107. Compare it to the expression you derived in Step 4.

$$F = \underline{\hspace{2cm}}$$

7. Apply the inputs shown in Table I and record the output state in the right-hand column.
8. Compare the two F output columns in Table I. What conclusion can you reach regarding the circuits in Figure 5-105 and 5-107?

Discussion

In this part of the experiment you demonstrated how TTL NAND and NOR gates are used to implement logic functions and how Boolean algebra is useful in minimizing the equation.

First we illustrated a standard logic circuit (Figure 5-104), and you wrote its output equation.

$$F = AC + AD + BC + BD$$

Then we illustrated how this circuit could be implemented with TTL NAND gates (Figure 5-105). It took four 2 input gates from the 74LS00 IC and one 4 input gate from the 74LS20, or two IC packages. Next you developed a truth table for this circuit.

In Step 4 you minimized the original equation with Boolean algebra. Your solution should look like this.

$$F = AC + AD + BC + BD$$

Factor out A, factor out B

$$F = A(C + D) + B(C + D)$$

Factor out (C + D)

$$F = (C + D)(A + B)$$

Then in Step 5 you constructed a circuit (Figure 5-107) made with a 74LS02 TTL NOR gate that implements the above reduced equation. Its output expression is $F = (A + B)(C + D)$. To verify its operation, you developed a truth table. By comparing the output results of the circuits in Figure 5-105 and 5-107 in Table I, you should find them identical. Obviously, the simpler circuit in Figure 5-107 is preferred because it will take up less space and will consume less power.

Procedure (continued)

In the following steps you will be given a Boolean equation to implement with CMOS NOR gates. You will implement the original expression and test it. Then you will minimize the expression and implement the reduced version. Finally, you will compare the logical operation of the two circuits.

9. Draw the AND and OR gate logic diagram of the expression

$$X = \bar{L}[\bar{K}(K + L) + M]$$
10. Redraw the circuit using positive NOR gates.
11. Implement your circuit in Step 10 with a 4001 CMOS quad two input NOR gate IC. The pin connections for the 4001 IC are given in Figure 5-108. Connect +5 volts to pin 14 and ground to pin 7. Connect all unused inputs to +5 volts.
12. Develop a truth table for the circuit. Use SW2, SW3 and SW4 to apply the K, L and M inputs. Monitor your output on L1. Record your output in the left hand X column of Table II.

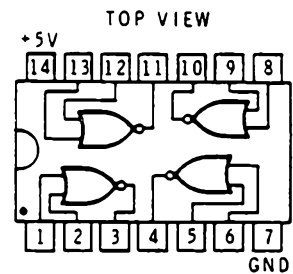


Figure 5-108
Pin Connections for 4001 CMOS IC.

INPUTS			OUTPUTS	
K (SW2)	L (SW3)	M (SW4)	X (Step 12)	X (Step 16)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table II

13. Reduce the expression in Step 9 using Boolean algebra. The minimized equation is.

$$X = \underline{\hspace{2cm}}$$
14. Draw the logic diagram of this circuit using AND and OR gates.
15. Implement the circuit developed in Step 14 with CMOS NOR gates.
16. Wire the minimized circuit and develop a truth table. Apply inputs K, L, and M with data switches SW2, SW3 and SW4. Monitor the output on L1. Use the right-hand X column in Table II to record your data.
17. Compare the two X output columns in Table II. What conclusions can you draw? What circuit minimization was really accomplished?

Discussion

Your logic diagram for the original expression in Step 9 should appear as in Figure 5-109. Redrawing the circuit using positive NOR gates should have given you the circuit in Figure 5-110. Your truth table should appear as in Table III.

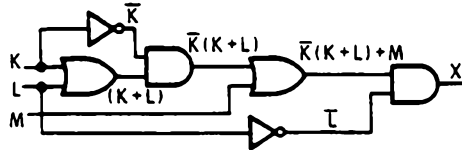


Figure 5-109

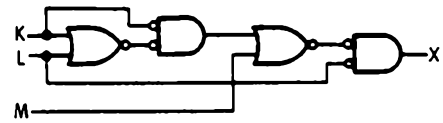


Figure 5-110

INPUTS			OUTPUT
K	L	M	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

TABLE III

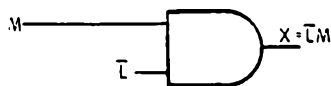


Figure 5-111



Figure 5-112

You next minimized the original expression by using Boolean algebra. Your reduction should appear like this.

$$\begin{aligned}
 X &= \bar{L}[\bar{K}(K+L) + M] \\
 X &= \bar{L}(\bar{K}K + \bar{K}L + M) \\
 X &= \bar{L}\bar{K}K + \bar{L}\bar{K}L + \bar{L}M \\
 X &= \bar{L}M
 \end{aligned}$$

A logic circuit for this is shown in Figure 5-111. This reduced expression is readily implemented with a positive NOR gate as shown in Figure 5-112. The K input has absolutely no effect on the circuit as this reduced expression indicates. The reduced circuit will produce the same logic function as the original more complex circuit as your truth table should indicate.

The important point to get from this exercise is that Boolean algebra accomplished a circuit minimization from 4 gates (Figure 5-110) to 1 gate (Figure 5-112). The type 4001 CMOS IC contains four 2-input NOR gates. It can be used to implement the original circuit (Figure 5-110). It is also used to implement the reduced circuit (Figure 5-112). But only one gate is used. The unused gates could possibly be used elsewhere.

Procedure (continued)

18. Write the truth table for a 3 input AND gate and sketch the appropriate logic symbol.
19. Show how to implement a three input AND gate with a quad 2 input NAND (7400). Draw the circuit, implement it on your Experimenter and verify its operation with a truth table.

Discussion

The truth table for a three input AND gate is given in Table IV. The logic diagram is shown in Figure 5-113.

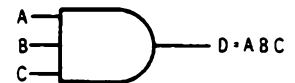


Figure 5-113

A 3-input AND gate implemented with a 7400 IC is shown in Figure 5-114. Inputs B and C are NANDed in gate 1. Gate 2 is connected as an inverter to produce \overline{BC} . \overline{BC} is then NANDed with A in gate 3 to produce $A(\overline{BC})$. Gate 4, connected as an inverter, gives $D = A(\overline{BC})$ which by the Law of Association is the same as $D = A B C$.

INPUTS			OUTPUT
A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Table IV

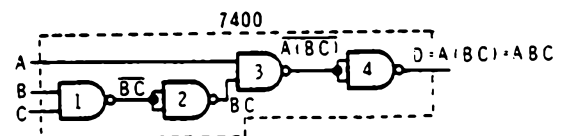


Figure 5-114

EXPERIMENT 8

The Wired – AND Connection

OBJECTIVE:

To investigate the wired-AND connection of an open-collector TTL logic gate and determine its logic function.

Introduction

With certain types of logic gates, connecting their outputs together as shown in Figure 5-115A, forms an additional logic function at the common connection. This logic function shown in Figure 5-115B is referred to as an implied, dot, or wired AND. NOR and AND functions can be achieved by simply tying gate outputs together to a common-collector pull-up resistor. This wired AND function is achieved by using open-collector TTL gates.

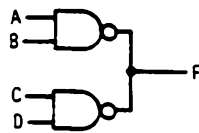


Figure 5-115A

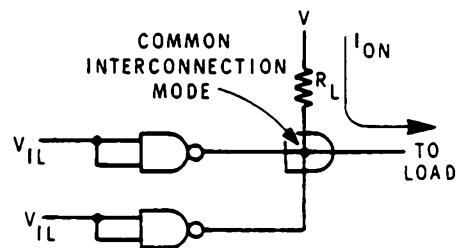


Figure 5-115B

Some logic families, when wired together as shown in Figure 5-115A, produce a wired OR condition. An example would be the ECL logic gate. There are two types of gates that will produce some very undesirable results when their outputs are connected together. These are logic circuits using active pull-up transistors such as the standard totem-pole TTL and CMOS.

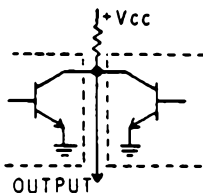


Figure 5-116

When gate outputs are connected together, the output transistor of each gate is connected in parallel. With open-collector TTL gate circuits, a single external pull-up resistor is added as shown in Figure 5-116. When either one or both of the shunt output transistors is conducting, the output will be low.

In this experiment, you will be investigating the effects of the wired-AND connection using an open collector TTL gate.

Materials Needed

Heathkit ET-3200 Digital Design Experimenter or equivalent.

- 1 — 74LS03 TTL IC (443-745)
- 1 — 74LS04 TTL IC (443-755)
- 1 — 74LS00 TTL IC (443-728)
- 1 — 74LS08 TTL IC (443-780)
- 1 — 1 k Ω resistor

Procedures

1. Mount the 74LS03 IC on the breadboard. Connect the circuit shown in Figure 5-117A. The pin connections for the 74LS03 are shown in Figure 5-117B. Connect +5 volts to pin 14 and the 1k resistor. Pin 7 is connected to GND. The inputs A and B will come from data switches SW1 and SW2. The output will be displayed on the LED indicator L1. L1 lit indicates a high output at C. L1 not lit indicates a low output at C.
2. Apply the inputs as indicated in Table V. Use positive logic. Record the corresponding outputs in column 1 of Table I.
3. Study your results in Table V. What logic function is being performed? _____
4. Write the output expression of this circuit from Table I using the procedure described in this unit. _____
5. Using DeMorgan's theorem, change its form. _____
6. Draw the equivalent logic diagrams for the expressions you derived in steps 4 and 5 above. Show the inputs and outputs.

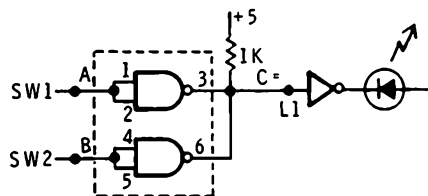


Figure 5-117A

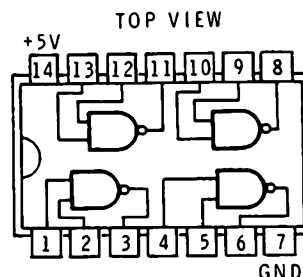


Figure 5-117B

Pin connections for
7403 integrated circuit.

7. Wire the circuit shown in Figure 5-118A. The pin connections for the 74LS08 and 74LS04 are shown in Figure 5-118B and C. Connect +5 volts to pin 14 and GND to pin 7 for both ICs.
8. Apply the inputs A and B from data switches SW1 and SW2 as shown in Table I. Use positive logic. Record the corresponding outputs in column 2 of Table V.
9. Compare columns 1 and 2 of Table V. Are they the same? _____
Why? _____

You can readily see that the AND function is being performed by the output AND gate in Figure 5-118A. What is causing the ANDING in Figure 5-117A? _____

What advantage does the circuit of Figure 5-117A have compared to the circuit of Figure 5-118A? _____

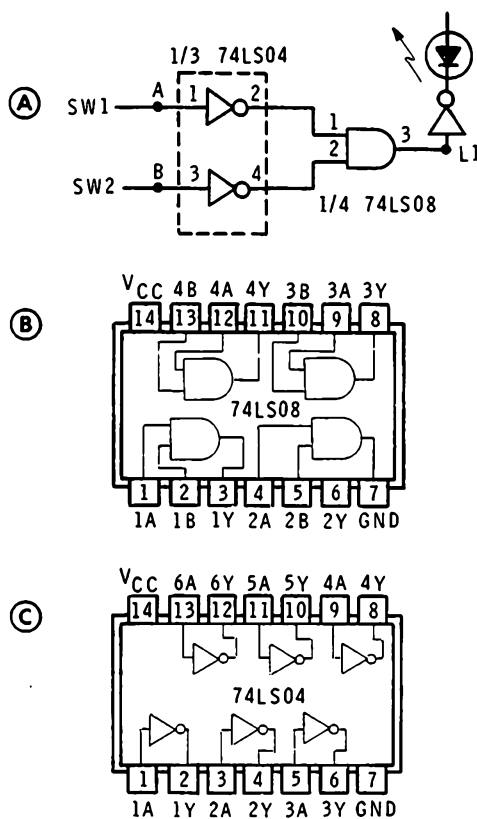


Figure 5-118

INPUTS		OUTPUT	OUTPUT
		Column 1	Column 2
$A_{(SW1)}$	$B_{(SW2)}$	$C_{(L1)}$	$C_{(L1)}$
0	0		
0	1		
1	0		
1	1		

Table V

10. Wire the circuit shown in Figure 5-119.
11. Apply the inputs A, B, C, and D from data switches SW1, SW2, SW3, and SW4 as indicated in Table VI. Use positive logic. Record the output state at L1 in column 1 of Table VI.
12. Study your results in Table VI. Notice that the output is a logic 0 when both inputs A and B are a logic 1 or; both inputs C and D are logic 1 or; when all inputs are logic 1.
13. From the data in Table VI, write the Boolean expression for the logic 0 outputs.

F = _____

14. Using Boolean algebra, reduce the equation to its simplest form. $F =$ _____. Note: Because your equation is derived from the logic 0 outputs, a vinculum should be placed over the entire expression after you have reduced it to its simplest form.
15. Draw the equivalent circuit(s) for the circuit of Figure 5-119 using standard TTL ICs. Show all inputs and outputs.

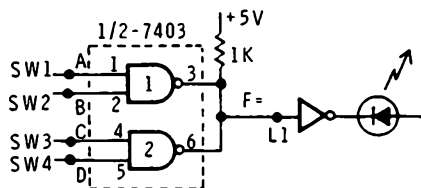


Figure 5-119

16. Wire the equivalent circuit of Figure 5-120A. The pin connections for the 74LS00 are shown in Figure 5-120B.

17. Apply the inputs indicated in Table VI. Record the corresponding outputs in column 2 of Table VI.

Do columns 1 and 2 match? _____ Why? _____

What are the advantages of the open collector circuit of Figure 5-119 compared to the standard TTL gate circuit of Figure 5-120? _____

INPUTS				OUTPUT 1	OUTPUT 2
SW1 A	SW2 B	SW3 C	SW4 D	OPEN COLLECTOR F	EQUIVALENT CIRCUIT F
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Table VI

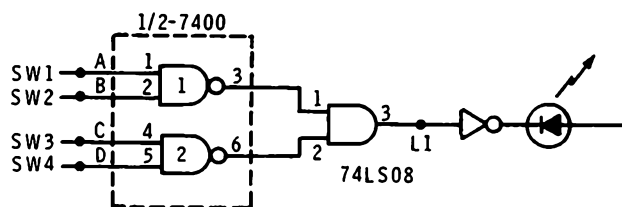


Figure 5-120A

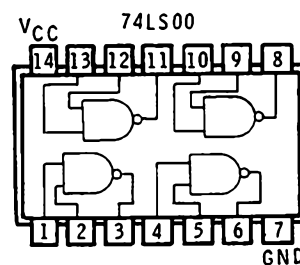


Figure 5-120B

Discussion

In step 1, you connected two of the gates in the 74LS03 as inverters with their outputs connected together through a 1 k Ω pull-up resistor. In step 2, you plotted a truth table for the circuit to see how it works. Then by studying the truth table, you should have concluded that the circuit performs the negative AND (Positive NOR) function since the output is 1 only when both of the inputs are logic 0. The output equation then is $C = \overline{A} \overline{B}$. In step 5, you used DeMorgan's theorem to change its form. You found that $C = \overline{A} \overline{B} = \overline{A + B}$. In step 6, you drew the equivalent circuits for these expressions. They should appear as shown in Figures 5-121A & B.

In step 7, you wired the equivalent circuit in Figure 5-118A. In step 8, you applied the inputs according to Table I. You recorded the corresponding outputs in column 2 of Table I. In step 9, you found that output columns 1 and 2 were the same. This shows that the circuits in Figures 5-117A and 5-118A are performing the same function. It is apparent that the AND function is being performed by the output AND gate in the equivalent negative AND circuit of Figure 5-118A. How is this being done in Figure 5-117A?

The two open-collector NAND gates in Figure 5-117A are connected as inverters. However, inverters of themselves can not perform ANDing functions. The only conclusion is that the ANDing is being done at the common connection point. This is the junction of outputs 3 and 6 of the NAND inverters and the 1k ohm pull-up resistor.

To further verify the wired AND connection, you wired the circuit in Figure 5-119. In step 11, you applied inputs according to Table VI. You recorded corresponding outputs in column 1 of Table VI. In step 13, you wrote the output expression F for the 0 outputs.

$F = A B \overline{C} \overline{D} + A B C \overline{D} + A B \overline{C} D + \overline{A} \overline{B} C D + A \overline{B} C D + \overline{A} B C D + A B C D$. With the use of Boolean algebra, you reduce the expression. Your reduction should be similar to the following:

The first step is to use the Laws of Commutation to regroup terms and factors.

$$A B \overline{C} \overline{D} + A B \overline{C} D + \overline{A} C D \overline{B} + \overline{A} C D B + A C D \overline{B} + A C D B + A B C \overline{D} = F$$

Next, factor out the common minterms. $A B \bar{C}$ in the first two terms; $\bar{A} C D$ in terms 3 & 4, $A C D$ in terms 5 & 6. Ignore term 7 for now.

$$A B \bar{C}(\bar{D} + D) + \bar{A} C D(\bar{B} + B) + A C D(\bar{B} + B) + A B C \bar{D}$$

By the Laws of Complements, this becomes:

$$A B \bar{C}(1) + \bar{A} C D(1) + A C D(1) + A B C \bar{D}$$

By the Law of Intersection, this becomes:

$$A B \bar{C} + \bar{A} C D + A C D + A B C \bar{D}$$

Rearranging terms by Commutation, we have:

$$A B \bar{C} + A B C \bar{D} + \bar{A} C D + A C D$$

Next, factor out common minterms; $A B$ in the first two terms, $C D$ in the last two terms.

$$A B (\bar{C} + C \bar{D}) + C D (\bar{A} + A)$$

Next, use the Law of Absorption on the first expression and the Law of Complements on the second. The result becomes:

$$A B (\bar{C} + \bar{D}) + C D(1)$$

Using DeMorgan's theorem on the first expression and the Law of Intersection on the second, you have:

$$A B (\overline{C D}) + C D$$

Using the Law of Absorption again, you have:

$$A B + C D$$

Because the original expression was derived from the logic 0 outputs, you must place a vinculum over the final expression. The final expression becomes:

$$\overline{A B + C D}$$

In step 15, you drew the equivalent circuits for the open-collector circuit of Figure 5-119. These circuits use the standard TTL ICs. They should appear as shown in Figures 5-121C and D. With the inputs shown, the output of each equivalent circuit matches the final result of the simplified expression above.

In step 16, you wired the circuit of Figure 5-120A to further verify that a standard TTL circuit must have an actual output AND gate to accomplish the same logic function as the open-collector wired AND.

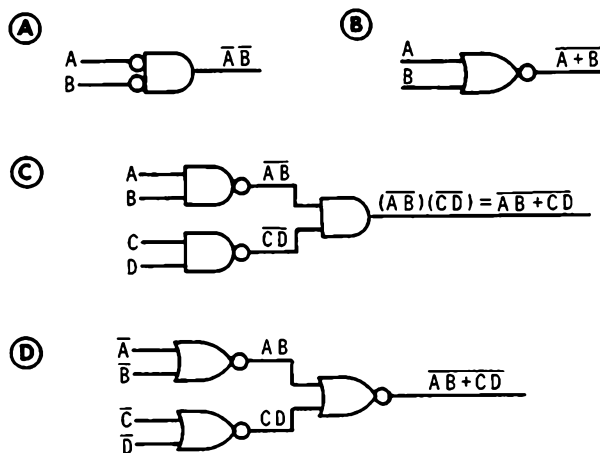


Figure 5-121

Standard TTL Equivalent Circuits.

In step 17, you applied the inputs indicated in Table II. You recorded the corresponding outputs in column 2 of Table II. A comparison of columns 1 & 2 indicates that both circuits function the same logically. They are equivalent but, the open-collector circuit of Figure 5-119 requires only one IC, the 74LS03. The obvious reason is that the common connection on the output is performing the AND function. In the standard TTL circuit an additional AND IC must be used on the output. Because fewer ICs are required in the open-collector circuit, it uses less power to accomplish the same result.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and work every problem first before you check the answer.

1. The mathematical system used to express, analyze, and design digital circuits is:
 - A. The binary number system.
 - B. Ordinary algebra.
 - C. Boolean algebra.
 - D. Canonical maps.
2. The Boolean equation for the circuit in Figure 5-122 is:
 - A. $W = \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + XY\bar{Z}$
 - B. $W = XY\bar{Z} + \bar{X}YZ + \bar{X}\bar{Y}Z$
 - C. $W = X\bar{Y}\bar{Z} + \bar{X}YZ + \bar{X}\bar{Y}Z$
 - D. $W = XYZ + \bar{X}YZ + \bar{X}\bar{Y}Z$

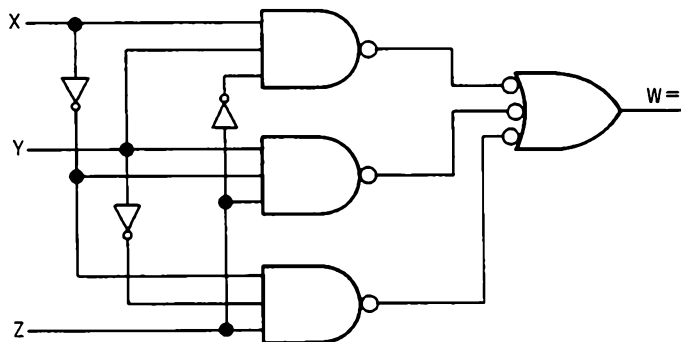


Figure 5-122

3. Given the expression, $A \bar{B} C + \bar{A} \bar{C} + \bar{B} D$, draw a circuit implementing standard AND & OR logic symbols. Then convert this circuit to an equivalent circuit using all positive logic NAND gates.
4. Given the expression $(A + \bar{B})(\bar{C} + \bar{D})$, draw a circuit implementing standard positive logic AND & OR symbols. Then convert this circuit to an equivalent circuit using all positive logic NOR gates.
5. The equation $\overline{A \bar{C} + \bar{A} \bar{B}}$ when minimized as much as possible becomes:
- $AC + \bar{A}B + \bar{B}C$
 - $\bar{A}C + A\bar{B}$
 - $AC + \bar{A}B$
 - $AC + \bar{A}B + BC$
6. The Boolean equation corresponding to the truth table below is:
- $W = XY\bar{Z} + X\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z$
 - $W = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + X\bar{Y}Z + XY\bar{Z}$
 - $W = \bar{X}YZ + \bar{X}\bar{Y}Z + XYZ$
 - $W = \bar{X}\bar{Y}Z + \bar{X}YZ + XY\bar{Z}$

INPUTS			OUTPUT
X	Y	Z	W
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

7. The equation $W = \bar{X}Y\bar{Z} + YZ + X\bar{Y} + Y\bar{Z}$ when minimized as much as possible by Boolean algebra, becomes:
- $W = X + Y$
 - $W = \bar{X} + \bar{Y}$
 - $W = Y\bar{Z} + YZ + X\bar{Y}$
 - $W = X\bar{Y} + \bar{Y}$

8. Which of the following are a form of DeMorgan's theorem?
- A. $\overline{A + B} = \overline{A} \overline{B}$
 - B. $\overline{A \overline{B}} = \overline{A + B}$
 - C. $\overline{A \overline{B}} = \overline{A + B}$
 - D. $\overline{A + B} = \overline{A} \overline{B}$
9. Using Boolean algebra and DeMorgan's theorem, the equation $D = (\overline{A} + \overline{B})\overline{C} + A \overline{C}$ can be minimized to:
- A. $A \overline{B} + \overline{A} \overline{C}$
 - B. $\overline{A} + \overline{C}$
 - C. $A + B \overline{C}$
 - D. $A \overline{C} + B \overline{C} + \overline{A}$
10. The Boolean equation of the wired AND circuit in Figure 5-123 is:
- A. $D = \overline{A}BC + A\overline{C} + B$
 - B. $D = (\overline{A}BC) + (A\overline{C}) + (\overline{B})$
 - C. $D = (\overline{A} \overline{C}) (\overline{A}BC) \overline{B}$
 - D. $D = (\overline{A} \overline{C}) (\overline{A}BC) (B)$

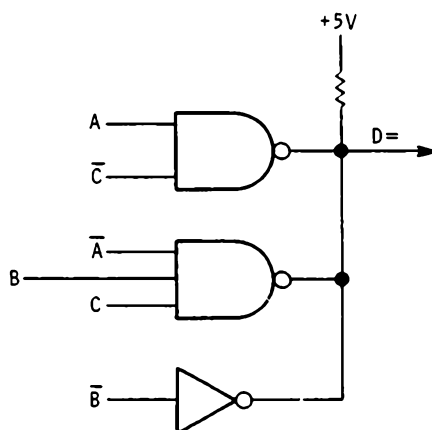


Figure 5-123

EXAMINATION ANSWERS

1. C — Boolean algebra
2. B — $W = XY\bar{Z} + \bar{X}YZ + \bar{X}\bar{Y}Z$
3. See Figure 5-124

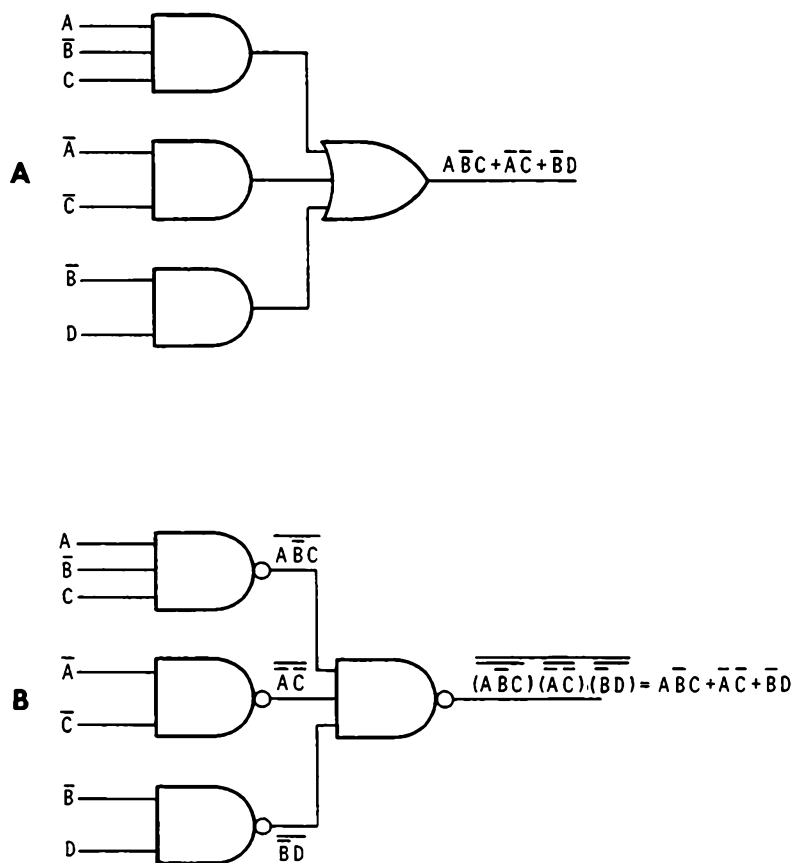


Figure 5-124

4. See Figure 5-125

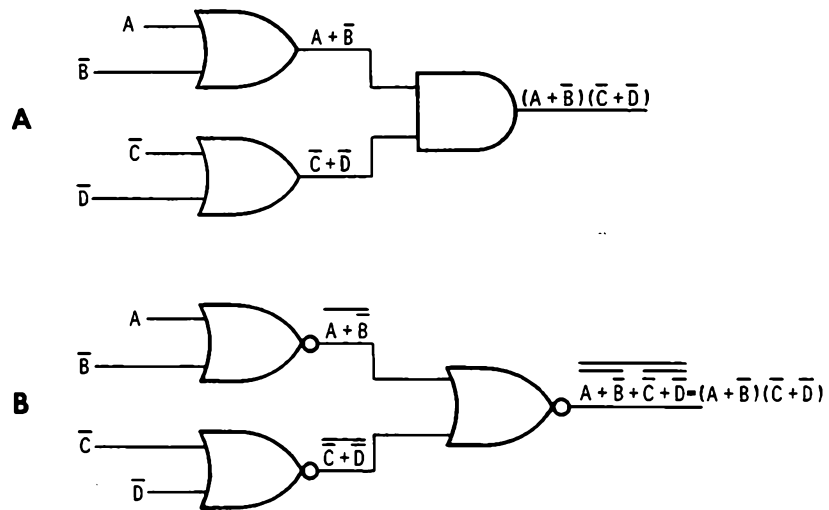


Figure 5-125

5. $C = AC + \bar{A}B$

DeMorgan's
DeMorgan's. Careful
with signs of grouping.

$$\overline{\overline{AC} + \overline{\bar{A}B}}$$

$$(\bar{A} + C)(A + B)$$

Law of Distribution
Law of Complements

$$\bar{A}A + \bar{A}B + AC + BC$$

$$\bar{A}B + AC + BC$$

Multiply the term BC
by 1 in the form of $A + \bar{A}$

$$\bar{A}B + AC + BC(A + \bar{A})$$

Law of Distribution

$$\bar{A}B + AC + ABC + \bar{A}BC$$

Law of Commutation
Law of Absorption

$$\bar{A}B + \bar{A}BC + AC + \bar{A}BC$$

$$\bar{A}B + AC$$

6. $D = W = \bar{X}\bar{Y}Z + \bar{X}YZ + XY\bar{Z}$

$$7. \quad A - W = X + Y$$

$$W = \overline{X}Y\overline{Z} + YZ + X\overline{Y} + Y\overline{Z}$$

Law of Commutation

$$W = \overline{X}Y\overline{Z} + Y\overline{Z} + YZ + X\overline{Y}$$

Law of Absorption $W = Y\overline{Z} + YZ + X\overline{Y}$

Factor out Y

$$W = Y(\overline{Z} + Z) + X\overline{Y}$$

Law of Complements

$$W = Y(1) + X\overline{Y}$$

Law of Intersection

$$W = Y + X\overline{Y}$$

Laws of Absorption and Commutation

$$W = X + Y$$

$$8. \quad B - \overline{A}\overline{B} = \overline{A + B}$$

$$D - \overline{A + B} = \overline{A}\overline{B}$$

$$9. \quad B - \overline{A} + \overline{C}$$

DeMorgan's theorem $D = (\overline{\overline{A} + \overline{B}})\overline{C} + \overline{AC}$

Expand $(AB)\overline{C}$ $D = (AB)\overline{C} + \overline{A} + \overline{C}$

Law of Absorption $D = AB\overline{C} + \overline{A} + \overline{C}$

$$D = \overline{A} + \overline{C}$$

$$10. \quad D = (\overline{AC})(\overline{ABC})(B)$$

APPENDIX—UNIT 5

SUMMARY OF BOOLEAN RULES

Here is a handy tabulation of all the commonly used Boolean algebra rules.

Laws of Intersection

$$A \cdot (1) = A$$

$$A \cdot (0) = 0$$

Laws of Union

$$B + 1 = 1$$

$$B + 0 = B$$

Laws of Tautology (Idempotent)

$$A \cdot A = A$$

$$B + B = B$$

Laws of Complements

$$A \cdot \bar{A} = 0$$

$$B + \bar{B} = 1$$

Law of the Double Negative

$$\bar{\bar{A}} = A$$

Laws of Commutation

$$A \cdot B = B \cdot A$$

$$A + B = B + A$$

Laws of Association

$$(A \cdot B)C = A(B \cdot C) = A \cdot B \cdot C$$

$$A + (B + C) = (A + B) + C = A + B + C$$

Laws of Distribution

$$AB + AC = A(B + C)$$

$$(A + B)(A + C) = A + BC$$

Laws of Absorption

$$A(A + B) = A$$

$$A + AB = A$$

$$\left[\begin{array}{l} A(\bar{A} + B) = AB \\ A + \bar{A}B = A + B \end{array} \right]$$

Common Identities**DeMorgan's Theorem**

$$\overline{AB} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A}\bar{B}$$

Unit 6

FLIP-FLOPS AND REGISTERS

CONTENTS

Introduction	6-3
Unit Objectives	6-4
Unit Activity Guide	6-5
RS Flip-Flops	6-6
D Type Flip-Flops	6-17
Storage Registers	6-20
JK Flip-Flops	6-26
Experiment 9 — Set-Reset Flip-Flop	6-37
Experiment 10 — D Type Flip-Flops	6-41
Experiment 11 — JK Flip-Flops	6-46
Unit Examination	6-51
Examination Answers	6-55

INTRODUCTION

In this unit you are going to learn about flip-flops. A flip-flop is a digital logic element used for storing one bit of binary data. It has two stable states, one representing a binary 1 and the other a binary 0.

The flip-flop is the basic logic element used in sequential logic circuits. The primary characteristics of a sequential circuit is memory. Such circuits are used for a variety of storage, counting, sequencing, and timing operations.

A major use of the flip-flop is in storage registers where a multibit binary word is stored. A register is made up of a number of flip-flops, each storing one bit of the number.

Read the Unit Objectives to see what you will learn, then follow the directions in the Unit Activity Guide. Be sure to record your time for each step.

UNIT OBJECTIVES

When you complete this unit you will have the following skills, knowledge, and capabilities:

1. You will be able to write a definition for a flip-flop.
2. You will be able to name the three basic types of flip-flops.
3. Given a logic diagram, you will be able to identify each of the three types of flip-flops from their symbols or logic gate connections.
4. You will be able to explain the operation of RS, D, and JK flip-flops, showing the output states for all possible input states.
5. Given a set of input waveforms for the RS, D, or JK flip-flop, you will be able to recognize the corresponding output waveforms.
6. You will be able to give a practical application for each of the three types of flip-flops.
7. You will be able to write a definition for a register.
8. Given a register made with any type of flip-flop, you will be able to measure the output states and determine the binary number stored there.

UNIT ACTIVITY GUIDE

**Completion
Time**

- | | |
|---|-------|
| <input type="checkbox"/> Read "RS Flip-Flops." | _____ |
| <input type="checkbox"/> Answer Self Test Review questions 1–11. | _____ |
| <input type="checkbox"/> Read "D Type Flip-Flops" | _____ |
| <input type="checkbox"/> Read "Storage Registers" | _____ |
| <input type="checkbox"/> Answer Self Test Review questions 12–16. | _____ |
| <input type="checkbox"/> Read "JK Flip-Flops." | _____ |
| <input type="checkbox"/> Answer Self Test Review questions 17–27. | _____ |
| <input type="checkbox"/> Perform Experiment 9. | _____ |
| <input type="checkbox"/> Perform Experiment 10. | _____ |
| <input type="checkbox"/> Perform Experiment 11 "JK Flip-Flops." | _____ |
| <input type="checkbox"/> Complete the Unit Examination. | _____ |
| <input type="checkbox"/> Review the Examination Answers. | _____ |

RS FLIP-FLOPS

A flip-flop is a digital logic circuit, whose basic function is memory, that is capable of storing a single bit of binary data. It can assume either of two stable states, one representing a binary 1 and the other a binary 0. If the flip-flop is put into one of its two stable states, it will remain there as long as power is applied or until it is changed.

A flip-flop remembers to which state it was previously set. It effectively memorizes the data it is given. You give it this data by applying appropriate logic inputs to it. To determine the value of the bit stored in the flip-flop, you look at its outputs.

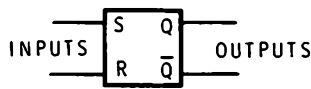


Figure 6-1
RS flip-flop
symbol

There are three basic types of flip-flops: the latch or RS, the D type, and the JK. Let's start with the simplest, the latch, also called a set-reset or RS flip-flop. This is the simplest form of binary storage element. The symbol shown in Figure 6-1 is used to represent this type of flip-flop.

The flip-flop has two inputs, S and R, and two outputs Q and \bar{Q} . Applying the appropriate logic signal to either the S or R input will put the latch into one state or the other. The S input is used to *set* the flip-flop. When a flip-flop is set, it is said to be storing a binary 1. The R input is used to reset the flip-flop. A reset flip-flop is said to be storing a binary 0.

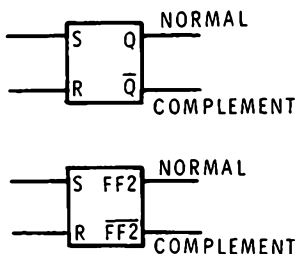


Figure 6-2

To determine which state the flip-flop is in, you look at the outputs. The latch has two outputs labeled Q and \bar{Q} . These are called the *normal* and *complement* outputs respectively. As in other logic circuits, any letter or alphanumeric mnemonic can be used to designate logic signals. See Figure 6-2. For example, the designation FF2 (meaning flip-flop number 2) could be used.

To tell what state the flip-flop is in, you look at the *normal* output. The logic level present there tells you which bit, 0 or 1, is being stored. If the normal output is a binary 0 level then the flip-flop is reset or storing a binary 0. If the normal output is the binary 1 level, the flip-flop is set and is storing a binary 1. The normal output always tells the state of the flip-flop. At the same time, the complement output has the state opposite that of the normal output. The complement output is just as useful in determining the output state of the flip-flop as long as you remember the above relationship. The simple table on the next page sums up the states of an RS flip-flop.

FLIP-FLOP STATE	OUTPUTS	
	Q	\bar{Q}
SET	1	0
RESET	0	1

This relationship is true for the latch and all other types of flip-flops.

A latch is readily constructed with logic gates as shown in Figure 6-3. Here two NAND gates are wired back-to-back so that the output of one feeds the input to the other.

Three methods of drawing the latch are illustrated in Figure 6-3. They are all electrically the same, but the version in Figure 6-3A is the most widely used. The other versions are used occasionally so it is a good idea to be familiar with the various configurations so that you will recognize them on a logic diagram when you see them.

The latch is also sometimes drawn using the negative logic NOR symbols. See Figure 6-4. Either NAND or NOR logic symbols can be used to illustrate a latch. Let's consider a latch using NAND gates.

Recall that if both inputs to a 2 input TTL NAND gate are binary 1, the output will be binary 0. In order for the output to go low, both (all) inputs must be binary 1 or high. Any other combination of inputs will produce a binary 1 (high) output. If both inputs are open, the output will also go low. For that reason, an open input has the same effect on the gate as a binary 1 input. The operation of the NAND gate is summarized by the truth table below.

INPUTS		OUTPUT
A	B	C
low	low	high
low	high	high
high	low	high
high	high	low

High usually refers to the most positive logic voltage level while low refers to the least positive level.

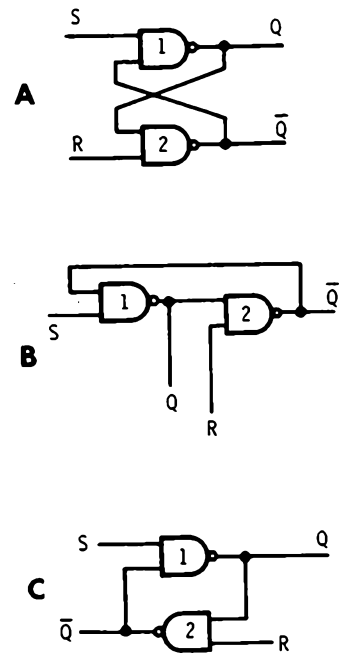


Figure 6-3

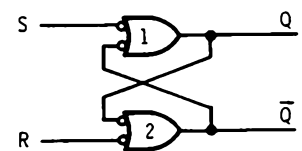


Figure 6-4

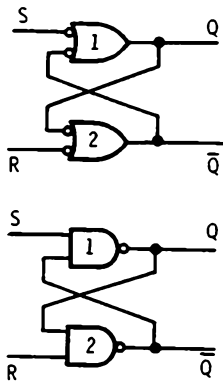


Figure 6-5

A look at the NAND gate truth table shows that of the four output states, three of them are high. The high output state is created by one or more low inputs. For that reason, we say that the predominant input state is a low or binary 0 for this type of gate. Now, let's consider the operation of the latch. Refer to Figure 6-5.

If the S and R inputs are both binary 1 (or open), which is the normal condition for this type of latch, the circuit is simply storing a bit put there by an earlier manipulation of the inputs. For example, if the flip-flop is set, the normal (Q) output from gate 1 (the set gate) will be high (binary 1). This output is fed back around to the upper input on gate 2 (the reset gate). The lower input to gate 2 is a binary 1 (or open) so its output \bar{Q} is low. The output from gate 2 is fed to the lower input of gate 1. This input holds the Q output high. You can see now why they call this circuit a latch. Because of this feedback arrangement, the flip-flop is latched into this state. It will stay this way until you change it. And the way you change it is by applying a low level to either the set or reset inputs.

If you apply a low to the R input of the latch, it will reset. The low level on the R input will force the output of gate 2 high. This will cause both inputs to gate 1 to be high so its output will go low thereby indicating the reset state.

If the flip-flop is set, applying a low to the S input will not do anything. The low level from the \bar{Q} output fed back to the set gate keeps the Q output high. In the same way, applying a low level to the R input while the latch is reset will not produce a state change.

So, summing it all up we can say that to set the latch you must apply a binary 0 to the S input. To reset it you must apply a binary 0 to the R input. The waveform timing diagrams in Figure 6-6 shows the effect of various inputs on the outputs.

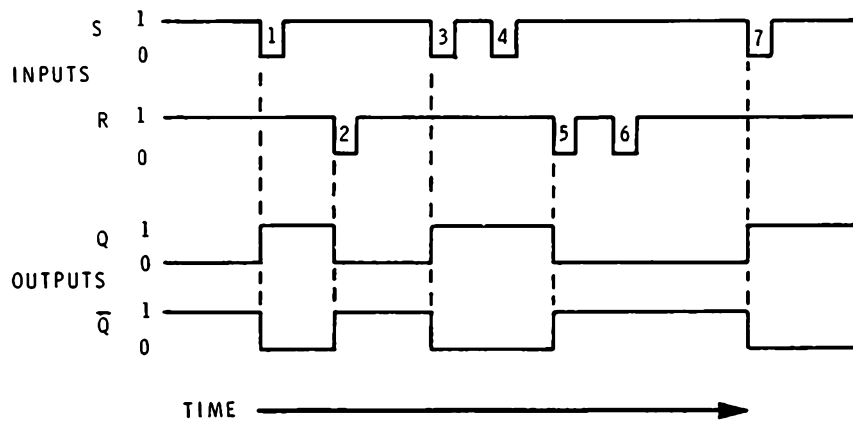


Figure 6-6

Go through these waveforms from left to right observing the effect of each input pulse on the outputs.

Looking at the waveforms above, you can determine what state the flip-flop is in prior to the application of pulse 1. Since the Q output is low and the \bar{Q} output is high, before the occurrence of (to the left of) pulse 1, the latch is in the reset state storing a binary 0. Now look at these waveforms again.

When pulse 1 occurs on the S input, the latch sets with the outputs going to their proper levels. Pulse 2 comes along next on the R input so the latch resets. Pulse 3 again sets the flip-flop. Note that pulse 4 also occurs on the S input. Since the flip-flop is already set naturally nothing will happen. Pulse 5 then resets the latch. Pulse 6, also occurring on the R input, has no effect on the state of the latch. Finally, pulse 7 again sets the latch.

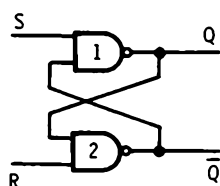


Figure 6-7

Both inputs should normally be high on a NAND gate latch unless you are changing its state. The high inputs do not disturb the state of the latch so either a binary 1 or binary 0 may be stored there. Short duration input pulses that switch from high to low should be used when the latch is to be set or reset.

Now look at the latch circuit in Figure 6-7. Let's see what happens if both inputs go low at the same time.

With both S and R inputs low, the Q and \bar{Q} outputs will *both* be high. No longer are the outputs complementary, therefore, we really don't know what state the flip-flop is in. It is in some ambiguous state and is neither set or reset. This condition is one of the peculiarities of a latch. When you are using it you have to be careful to avoid simultaneous low inputs on the S and R terminals. This ambiguous state is generally undesirable because it can produce undesirable operation of a logic circuit if it is not avoided or accounted for. The ambiguous condition actually represents a third state in which the latch can exist. This state is sometimes referred to as the "limbo" state.

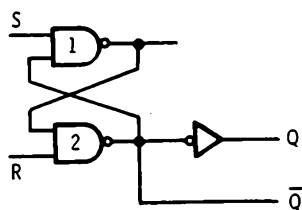


Figure 6-8

One way to avoid this condition is to modify the latch as shown in Figure 6-8. The normal and complement outputs are both derived from gate 2. The inverter ensures that the outputs are always complementary even if both inputs do go low simultaneously.

The complete operation of a NAND gate latch is summarized by the truth table below.

INPUTS		OUTPUTS		STATE
S	R	Q	\bar{Q}	
0	0	1	1	Limbo
0	1	1	0	Set
1	0	0	1	Reset
1	1	X	\bar{X}	Either set or reset

The truth table accounts for all possible input and output states. Note that when both S and R inputs are binary 1, the output state of the flip-flop is designated X, where X can be either a 0 or a 1 as determined by previous input conditions.

The latches we've discussed so far use positive logic NAND gates. We can also make latches out of positive logic NOR gates. Such a latch is shown in Figure 6-9.

It is identical to the other latches just discussed in that the two gates are wired back-to-back. Even the logic symbol is the same. But first, refresh your memory on positive NOR gate operation by referring to the truth table below.

INPUTS		OUTPUTS
A	B	C
low	low	high
low	high	low
high	low	low
high	high	low

A high or binary 1 on either or both inputs produces a low or binary 0 output. This is considerably different from the NAND gate so the effect is to make the operation of a NOR latch completely different from the NAND latch.

While NOR and NAND latches have exactly the same function, they achieve it in a slightly different way. To set the NOR latch you apply a binary 1 to the S input. To reset it you apply a binary 1 to the R input. Normally, both inputs should reset at binary 0. And, if both inputs are made binary 1 simultaneously, the "limbo" or ambiguous state occurs. This is the exact opposite set of input conditions that exist in the NAND latch. Take a look at the circuit in Figure 6-10.

Note a subtle difference. The R and S inputs are reversed from those on the NAND latch. The reason for this has to do with the characteristics of the NOR gate. Applying a binary 1 to the R input forces the output of gate 1 low. This makes the upper and lower inputs to gate 2 low or binary 0 so its output is a binary 1. With this arrangement ($Q = 0$, $\bar{Q} = 1$) the flip-flop is clearly reset. As you can see, the interpretation of the outputs is the same. In fact, it is the same for any flip-flop.

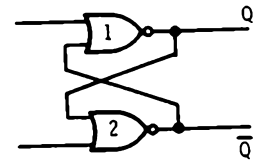


Figure 6-9

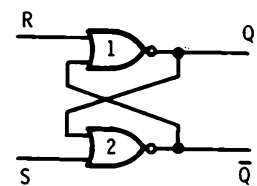


Figure 6-10

Both outputs will go to the binary 0 state if a binary 1 level is applied to the R and S inputs simultaneously. This is exactly the opposite of what happens in the NAND latch. Nevertheless, this ambiguous condition is generally avoided unless there is some specific application for it.

The operation of the NOR latch Figure 6-10 is summarized in the truth table below.

INPUTS		OUTPUTS		
S	R	Q	\bar{Q}	STATE
0	0	X	\bar{X}	Either set or reset
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Ambiguous

As in the NAND latch, the X output indicates either set or reset.

One of the most common and useful applications for a latch flip-flop is in switch buffering. Pushbutton switches are used in digital equipment to control various aspects of its operation. However, most pushbutton switches produce contact bounce. When the button is depressed or released, the switch contacts do not make an immediate solid electrical or mechanical connection. The contacts "bounce" open and closed for a brief period of time.

The waveform in Figure 6-11 indicates this effect.

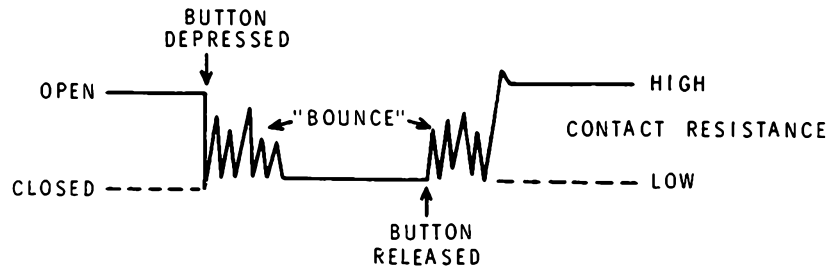


Figure 6-11

This waveform could represent contact resistance. Naturally, if current is being switched, this waveform would represent the voltage across the switch. Instead of getting solid off-on switching, you get pulses. Such pulses can repeatedly trigger digital circuits. In pressing the button once, you would expect to get a single pulse or level change. Instead, the contact bounce gives you several. This effect is usually detrimental to the performance of digital circuits.

The circuits in Figure 6-12 show two ways of using a pushbutton switch to supply a logic pulse or level change. Such circuits usually produce a considerable amount of contact bounce.

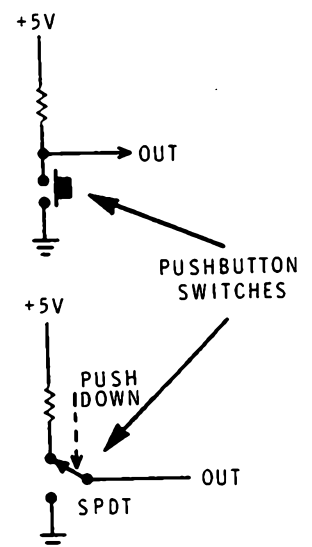


Figure 6-12

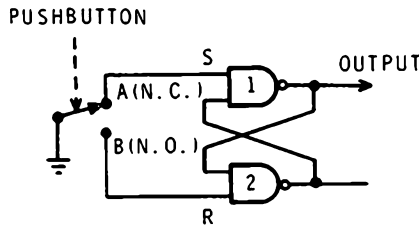


Figure 6-13

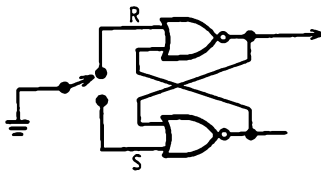


Figure 6-14

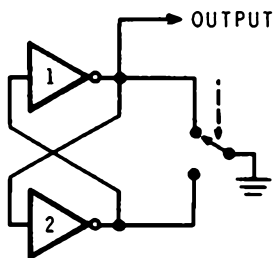


Figure 6-15

To overcome this problem, the switch can be combined with a latch as shown in Figure 6-13. An SPDT break-before-make (non-shorting) momentary contact pushbutton switch is normally used. With the switch in position A (not depressed or normally closed, N.C.), the output of gate 1 is held high. Depressing the switch, so that the grounded arm contacts position B, forces the output of gate 2 high and the output of gate 1 low. So how is the “bounce” removed? Well, as the button is depressed the arm of the switch breaks contact with point A. Even though it may bounce several times between open and ground, it has no effect on the state of the latch. The effect is the same as trying to repeatedly set a latch that is already set. Nothing happens. As the contact arm is in transit between points A and B, both inputs to the latch are open so the latch simply remains set. As point B is contacted, the latch resets. The slightest disturbance will trigger the state change through quickly. Even if switch bounce occurs, the latch is insensitive to it. The result is a single clean logic level change at the output. Releasing the pushbutton causes the latch to change back to its original state.

A NOR latch can also be used to buffer contact bounce. This is illustrated in Figure 6-14. The NOR latch removes contact bounce just as well as the NAND latch, only the reference R and S need be reversed on the NOR latch.

Another switch buffer latch circuit is shown in Figure 6-15. The latch is made of inverters so the outputs and inputs are common. The switch normally holds the output of inverter 1 low so that inverter 2 output is high. Pressing the switch reverses this state. The output is a “bounceless” level change.

The most important requirement for a switch used with a latch buffer is that it must have two terminals, so an SPDT unit is required. And it is important that it be a break-before-make type so that the A and B contact points are not shorted momentarily in switching from one position to the other. This would put the latch into its “limbo” state briefly and false triggering will occur.

Another point is that the switch doesn’t necessarily have to be a momentary contact type pushbutton. Any SPDT switch, slide or toggle, can be used. If it supplies logic level changes to a digital circuit, it will probably need buffering.

Self Test Review

1. What will the normal output level of a latch be if it is set?
 - a. high
 - b. low
2. The complement output of a latch is low. What is the bit value stored?
 - a. binary 0
 - b. binary 1
3. Normally the duration of the pulses applied to the set or reset inputs should only be long enough to put the latch in the proper state.
 - a. True
 - b. False
4. Which of the following is *not* a typical name for the circuit discussed in this section?
 - a. latch
 - b. RS flip-flop
 - c. set-reset flip-flop
 - d. multivibrator
5. The ambiguous state in a latch is indicated by which of the following conditions?
 - a. both outputs low
 - b. both outputs high
 - c. either a or b
 - d. one output low, the other high
6. Unless the state of a NAND gate latch is being changed its inputs should both be
 - a. high
 - b. low
 - c. open
7. Both inputs of a NAND latch are low. The state of the latch is:
 - a. set
 - b. reset
 - c. ambiguous
8. Both inputs to a NAND latch are low. The S input goes high. Shortly thereafter, the R input goes high. The state of the latch is
 - a. set
 - b. reset
 - c. ambiguous

9. Both inputs to a NOR latch are high. The R input goes low, then the S input goes low. What is the value of the bit stored in the latch?
 - a. binary 0
 - b. binary 1
10. Besides the storage of binary data, latches are also commonly used for _____.

Answers

1. a. high
2. b. binary 1
3. a. True
4. d. multivibrator
5. c. Either a. or b. The ambiguous state is indicated by two high outputs in a NAND latch and two low outputs in a NOR latch.
6. a. high
7. c. ambiguous
8. b. reset. The *last* or most recent input level determines the state of the latch.
9. b. binary 1. See explanation in 8 above.
10. switch buffering (to eliminate contact bounce).

D TYPE FLIP-FLOPS

Now let's consider the D type flip-flop. Its symbol is shown in Figure 6-16.

Like any other flip-flop the D flip-flop has two outputs that are used to determine its contents. That is, the outputs indicate what bit is stored there. The Q output tells you the state of the flip-flop directly. If it is a binary 0, the flip-flop is reset. If it is a binary 1, the flip-flop is set.

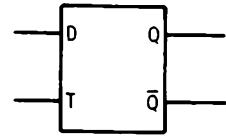


Figure 6-16

Now look at the inputs. Like on the latch there are two. But they work differently. The D input is where you apply the data or bit to be stored. Of course, it can be either a binary 1 or a binary 0. The T input line controls the flip-flop. It is used to determine whether the input data is recognized or ignored. If the T input line is high or binary 1, the data on the D line is stored in the flip-flop. As long as the T line is high, the normal output will simply follow or track the D input. If the T line is low or binary 0, the D input line is not recognized. The bit stored in the flip-flop previously is retained. The D line can essentially do anything and it will just be ignored if T is low.

You can get a better idea about how the D flip-flop works by taking a look at its insides. The logic diagram of one type of D flip-flop is shown in Figure 6-17.

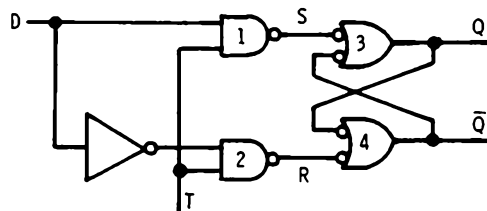


Figure 6-17

Gates 3 and 4 form a latch where the bit is stored. Gates 1 and 2 are enabling gates that pass or inhibit the input. The inverter makes sure that the S and R inputs to the latch are always complementary to avoid any possibility of the ambiguous state occurring.

With a low input on the T line, the outputs of gates 1 and 2 are high. This is the normal state for the inputs of a NAND latch to assume. In this state the latch is undisturbed.

Refer to Figure 6-17. Suppose a binary 1 is applied to the D input. Of course, nothing happens if the T input is still low. Now, make the T input go high. This enables both gates 1 and 2. The binary 1 on the D input makes gate 1 output go low. The inverter puts a low on the input to gate 2 so its output stays high. The low output of gate 1 sets the latch causing it to store the binary 1. Returning the T input low disables the input, but the binary 1 is retained.

Now look at the waveforms in Figure 6-18. These represent the D and T inputs and the Q output of a D flip-flop. The output is identical to the D input as long as the T input is high. When the T line goes low, the flip-flop stores the *last* state it sees on the D input.

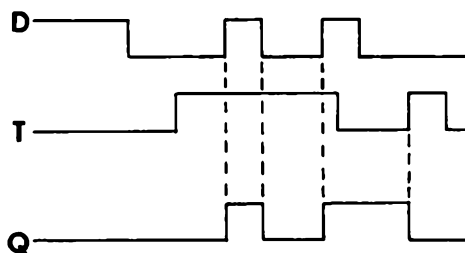


Figure 6-18

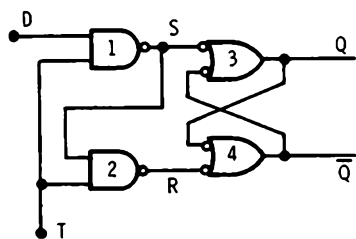


Figure 6-19

The circuit in Figure 6-19 shows another method of implementing the D flip-flop. As in the other circuit, gates 3 and 4 form the storage latch while gates 1 and 2 handle the input control.

Note that no separate inverter is needed. This arrangement functions exactly like the other circuit but is more economical of logic circuits. This circuit is quickly and easily made from a common quad 2 input NAND IC.

The operation of a D type flip-flop is completely described by the truth table below.

INPUTS		OUTPUTS	
D	T	Q	\bar{Q}
0	0	X	\bar{X}
0	1	0	1
1	0	X	\bar{X}
1	1	1	0

Note that when T is binary 1, the Q output is the same as the D input. When T is binary 0, the Q output can be either binary 0 or 1 depending upon a previous input. This is indicated by the X state in the table. Note that a D flip-flop does not have an ambiguous state.

A D flip-flop can also be constructed with positive NOR gates as indicated in Figure 6-20.

To explain the NOR latch better, however, it is desirable to redraw it so that the gates are shown as they are used. See Figure 6-21.

Gates 3 and 4 make up the latch that can be set or reset by the inputs from gates 1 and 2. Gates 1 and 2 control the input in that they determine whether the D input will be transferred to the latch. Functionally they perform the *AND* operation.

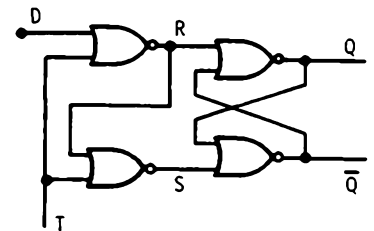


Figure 6-20

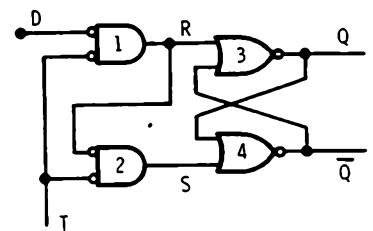


Figure 6-21

The NOR flip-flop does not perform exactly like the NAND flip-flop, but it is similar. Both circuits store one bit of information. The recognition of the D input is determined by the state of the T input. And, here lies the difference. In the NAND flip-flop, the T line has to be high in order for the flip-flop to store the D input state. In the NOR flip-flop, the T line must go *low* in order to recognize the D input. Bringing the T line high disables the D input. The last D input state prior to the T input going high is stored.

STORAGE REGISTERS

The most common application of the D flip-flop is as an element in a storage register. A register is a group of flip-flops used to store a binary word. Each flip-flop stores one bit of the data word. For example, a word could consist of four bits. To store this word, we need one D flip-flop for each bit. The result is a 4-bit storage register.

Figure 6-22 below illustrates a 4-bit register.

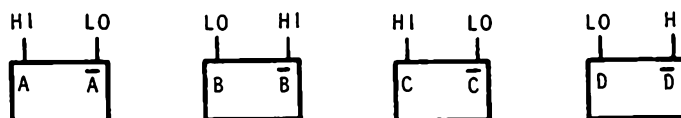


Figure 6-22

Each flip-flop is labeled with its own designation, A through D, so that it can be identified. Also shown here are the states of the flip-flop.

With the information given in Figure 6-22, you really can't tell what number is stored. True, you can look at normal outputs of the flip-flop and write down the corresponding bit values. This will give you two possible bit patterns depending upon whether you read from right to left or left to right. These bit patterns are 1010 (left to right) and 0101 (right to left).

There is one missing ingredient. Which bit is the most significant (MSB)? The answer is, it could be A or it could be D. Usually the LSB is designated as the earliest letter of the alphabet or the lowest number if the number designations are used. If A is the LSB above then the number stored there is 0101 or a decimal 5. If D is the LSB the number is 1010 or a decimal 10. It's always necessary to identify the LSB and/or MSB positions on a register in a logic diagram.

To be sure you understand this, let's take another example. A five-bit register with flip-flops A, B, C, D, and E is storing a number. The flip-flop states are: A-reset, B-set, C-set, D-reset, and E-set. If the A flip-flop is the LSB, the binary number is $\bar{E}DCB\bar{A}$ or 10110. This converts to a decimal 22.

Another type of register you should be familiar with is the switch register. This is exactly what its name implies, a register made of switches. The group of switches form a register for storing a single word or bit pattern using one switch per bit. The position of the switch (up/down, on/off, open/closed, etc.) determines the bit value.

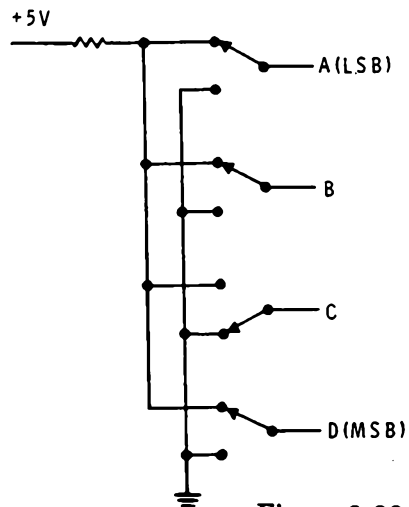


Figure 6-23

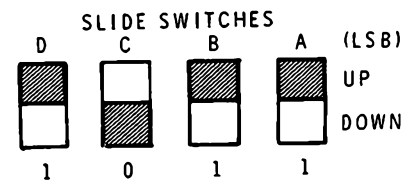


Figure 6-24
A 4-bit switch register
storing the number 1011.

A 4-bit switch register is shown in Figure 6-23. Four single pole double throw (SPDT) switches store the bit value as a physical position. The A, B, C, and D outputs are ground (0) or +5 (1) depending upon their position. The binary number stored in the switch register is 1011 or decimal 11. You can determine the switch register contents by monitoring its electrical outputs. Or in most switch registers, the switches are mounted adjacent to one another horizontally with the LSB on the right and their position (usually up or down) is readily observable. Up usually means 1, down means 0. Therefore, a visual identification of the switch register contents is possible. See Figure 6-24.

A frequent operation in digital equipment is the transfer of data from one register to another. Figure 6-25 illustrates how the data in a switch register can be transferred to a register made of D flip-flops. The switch outputs are fed to the D inputs. The control of the transfer is by the common T line on the flip-flops.

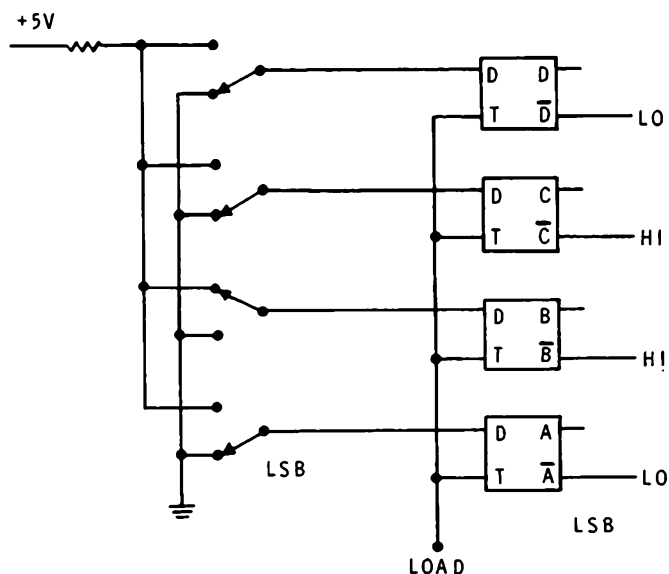


Figure 6-25

The content of the flip-flop register in Figure 6-25 is 1001. The data given in the figures is the complement output states. From this, you can determine the normal outputs and hence the contents of the register. You can also determine the output of the switch register, 0010, by inspection. With the T inputs to the flip-flop at binary 0, the data input from the switch register is not recognized by the flip-flops. But if the LOAD control line goes high momentarily, the flip-flop register contents will become the same as the switch register, 0010.

There are two important points to note here. First, the LOAD input controls the transfer of the data from the switch register to the flip-flop register. This LOAD input is the parallel or simultaneous control of all the flip-flop T inputs. This line is also sometimes called the STROBE input since it is usually only enabled or “strobed” momentarily with a binary 1 pulse to transfer the data.

Second, the data transfer is a parallel one. That is all bits from the switch register are loaded into the flip-flop register simultaneously.

Instead of drawing the individual flip-flops, most registers are shown as only a single box with the inputs and outputs identified as shown in Figure 6-26.

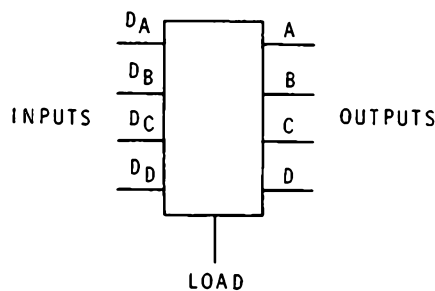


Figure 6-26

This is particularly true of MSI integrated circuit registers. Many IC registers also do not have the complement outputs available.

Self Test Review

12. The T input of a D type flip-flop determines its state.

- True
- False

13. D type flip-flops are widely used to form _____

14. Complete the truth table of a NOR gate D flip-flop.

INPUTS		OUTPUTS	
D	T	Q	\bar{Q}
0	0		
0	1		
1	0		
1	1		

15. What is the decimal equivalent output of the register shown in Figure 6-27? Assume positive logic. _____

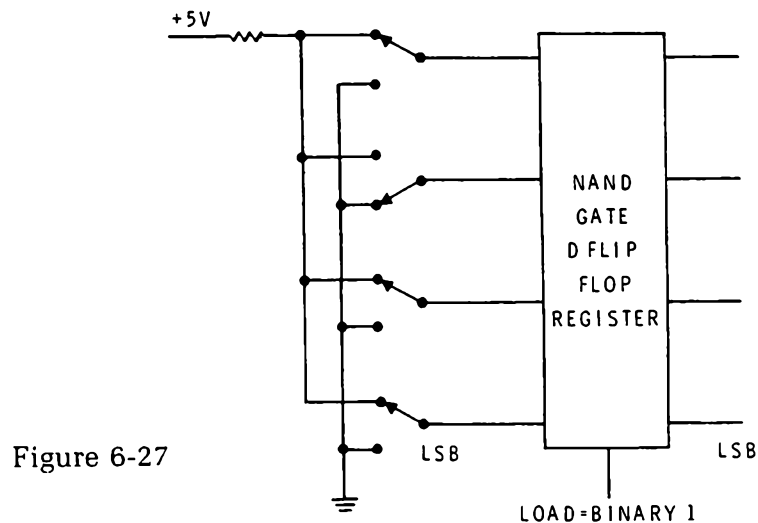


Figure 6-27

16. Given the input waveforms shown in Figure 6-28, sketch the normal output waveform of a NAND gate D type flip-flop.

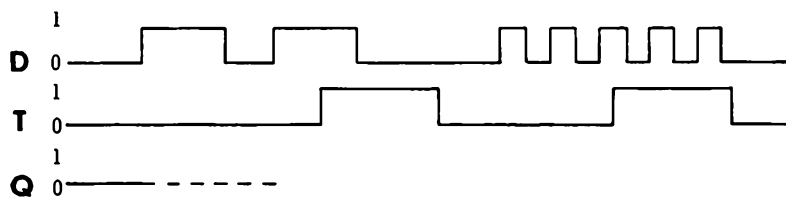


Figure 6-28

Answers

12. False
 13. storage registers
 14.

INPUTS		OUTPUTS	
D	T	Q	\bar{Q}
0	0	0	1
0	1	X	\bar{X}
1	0	1	0
1	1	X	\bar{X}

$X = 0 \text{ or } 1$

The normal (Q) output will be the same as the D input when T is low. When T is high, the D input is ignored, and the flip-flop simply retains the bit X stored there previously.

15. 11_{10} The output of the switch register is 1011. This is stored in the flip-flop register since the LOAD line is high. Therefore, the register output is $1011_2 = 11_{10}$.
 16. See Figure 6-29.

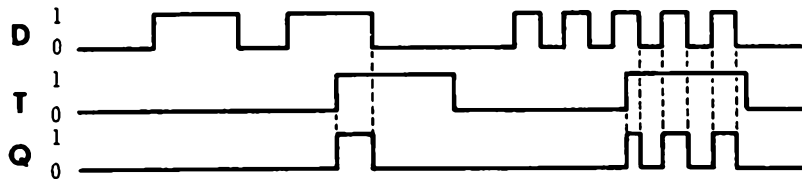


Figure 6-29

JK FLIP-FLOPS

The JK flip-flop is the most versatile type of binary storage element in common use. It can perform all of the functions of the RS and D type flip-flops described earlier plus it can do several other things that these simple flip-flops cannot. Naturally, it is more complex and expensive than the other types so for that reason it isn't always used where simpler and less expensive circuits will do.

An integrated circuit JK flip-flop is really two flip-flops in one. It usually consists of two latches, one feeding the other, with appropriate input gating on each. See Figure 6-30.

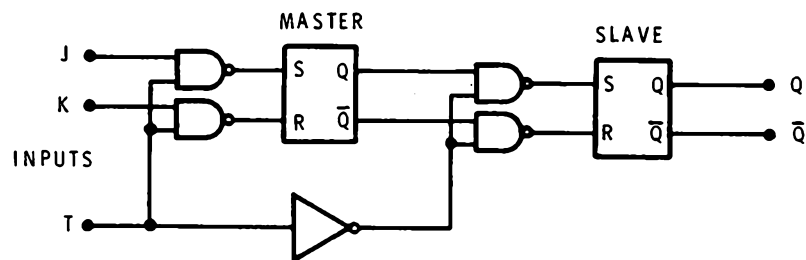


Figure 6-30

The arrangement is called a master-slave JK flip-flop. The master flip-flop is the input circuit. Logic signals applied to the JK flip-flop set or reset this master latch. The slave flip-flop is the latch from which the outputs are taken. The slave latch gets its input from the master latch. Both latches are controlled by a clock pulse. Since there are *two places* to store bits in a JK flip-flop, there are times when both master and slave latches are identical or times they are complementary. But only one of these latches is responsible for indicating the state of the JK flip-flop. The slave latch designates the state being stored. If it is set, the JK flip-flop is storing a binary 1.

Refer to Figure 6-31.

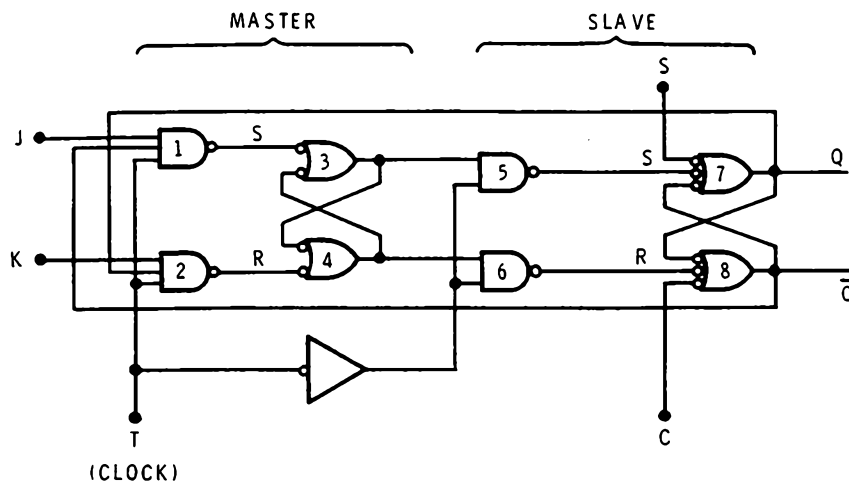


Figure 6-31

The logic gates are the positive NAND type. Gates 3 and 4 make up the master latch while its input is controlled by gates 1 and 2. The slave latch is made up of gates 7 and 8. Gates 5 and 6 control the transfer of the master latch state to the slave latch. Note that clock signal T controls the input gating circuits. The inverter keeps the clock to the master and slave input gates complementary. The clock pulse controls the JK flip-flop while the J and K inputs determine exactly how it will be controlled. (Note: You will also see the T input referred to as C or CLK designating clock pulse or clock.)

The set (S) and clear (C) inputs also control the JK flip-flop. These are inputs to the slave latch that can be used to set or reset (clear) the flip-flop. These inputs override all other circuitry in the JK flip-flop. These inputs are used to preset the state of the flip-flop prior to any other operation involving the JK inputs and the clock. They work just like the inputs on any latch.

To set the slave latch *and* the JK flip-flop, the S input should be low, the C input high. This forces the normal Q output high indicating that a binary 1 is being stored. To reset the JK flip-flop, the C input is made low while S is high. Normally, the S and C inputs will be high when they are not being used to preset the flip-flop. This arrangement is identical to that for the NAND latch.

Now let's consider how the J, K, and T (clock) inputs affect the flip-flop. Refer to Figure 6-31. Consider the time when the clock input is low. Gates 1 and 2 will be inhibited so the J and K inputs cannot control the state of the master latch. The master latch can be in either state. At this time, the slave latch will have the same state as the master latch when the clock input is low. The output of the inverter in the clock line is binary 1, causing the gates 5 and 6 to be enabled during this time. Therefore, the state of the master latch is simply transferred to the slave latch. For example, if binary 1 is stored in the master latch, the output of gate 3 will be high and the output of gate 4 will be low. This will make the output of gate 5 low and the output of gate 6 high. This low on the input to gate 7 will force its output high, thereby setting the slave latch and storing a binary 1.

Now if the clock T goes high, gates 1 and 2 will be enabled. The output of the inverter will inhibit gates 5 and 6. The master latch cannot further change the slave latch. But now with gates 1 and 2 enabled, the J and K inputs can affect the state of the master latch. The JK flip-flop outputs Q and \bar{Q} are fed back around to gates 1 and 2 where, along with the J and K inputs, they will also determine the state of the master latch.

If both J and K inputs are low, the outputs of gates 1 and 2 will be held high, so no change takes place in the master latch.

If the J and K inputs are both high (or open), then the state of the master latch will be determined by the Q and \bar{Q} outputs. For example, if the slave latch is set, the master latch will be reset. If the slave is reset, the master will be set. The reason for this is the way the outputs are crisscrossed back to gates 1 and 2. Remember that with the J, K, and T inputs high, the state of the master latch will be determined by the Q and \bar{Q} outputs.

Now let's consider the effect of the J and K inputs. These inputs are analogous to the set and reset inputs on a latch. If J is 1 and K is 0, we will set the master latch. If J is 0 and K is 1, the master latch will be reset. Remember, the T input line must be high for this to happen.

The state of the JK flip-flop is the state of the slave latch. The state of the slave latch is determined by the master latch. The state of the master is, in turn, determined by the J and K inputs. And to top it off, the clock input determines *when* each of these latches will be affected. With the clock input high, only the master latch will be affected. The inverter on the clock line blocks gates 5 and 6 so the slave latch is not disturbed. The states of the JK inputs will ultimately determine the output state but only at a specific time. When the clock line switches from high to low (trailing edge), the state of the master latch is transferred to the slave latch.

When the clock (T) is high, gates 1 and 2 will be enabled therefore the master latch will be changed by either the Q and \bar{Q} outputs or by the J and K inputs. When the clock goes low, the state of the master latch is transferred to the slave latch through gates 5 and 6, which are enabled at this time. Gates 1 and 2 are inhibited and the J and K inputs have no effect.

If the JK inputs are high or open, the flip-flop will change state each time the clock input switches from high to low. To show this, assume that the clock input is high, the JK inputs are high, and the slave latch is set. The Q and \bar{Q} lines fed back to gates 1 and 2 cause the master latch to be reset. Then when the trailing edge of the clock pulse occurs, (clock switches from 1 to 0), the reset state in the master is transferred to the slave. The JK flip-flop is now reset. When the clock again goes high, the slave latch then sets the master latch. As the clock goes low, the set state in the master is transferred to the slave. As you can see then, with the JK inputs high, the flip-flop complements itself each time the clock switches from high to low (trailing edge). We call this operation toggling.

Assume the J and K inputs are open or high. The waveform in Figure 6-32 represents the normal flip-flop output.

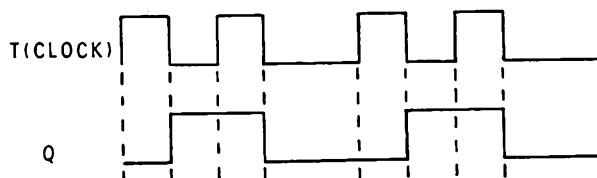


Figure 6-32

When the clock switches from 1 to 0, the state of the flip-flop changes. The output is not affected during the 0 to 1 transition (leading edge).

If you will look at the clock and output waveforms in Figure 6-33, you will see a definite relationship. The Q output has a frequency one half the T input. The reason for this is simply that the flip-flop changes state on only the trailing edge or every other transition of the clock. Therefore, the JK flip-flop in its toggling mode is a two to one frequency divider. It halves any input frequency applied to the clock input. If an input of 50 KHz is applied, the output will be one half or 25 KHz. Cascading JK flip-flops permits frequency division by any factor of 2 (2, 4, 8, 16, 32, 64, etc.). The frequency division ratio is 2^n where n is the number of flip-flops cascaded.

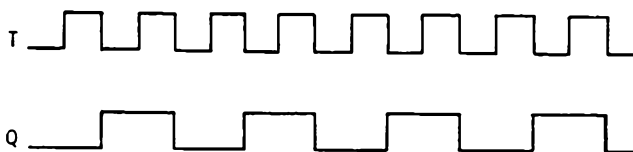


Figure 6-33

We have now considered all of the modes of operation of the JK flip-flop, but let's review them each briefly. Refer to Figure 6-31.

First, there are the S and C inputs. The effect of these can be summed up by the truth table below.

INPUTS		OUTPUTS	
S	C	Q	\bar{Q}
1	1	X	\bar{X}
0	1	1	0
1	0	0	1
0	0	1	1

This is exactly the same truth table we established for the NAND latch. Yes, the JK flip-flop does have an ambiguous state. If both S and C inputs are low, both the Q and \bar{Q} outputs will be high. Therefore, care should be taken to see that this condition does not occur.

The set (S) and clear (C) inputs are used to preset the flip-flop to some desirable condition prior to another operation. The most common operation is to reset it. For that reason many IC JK flip-flops have only a C input line. Use of the S and C inputs is referred to as asynchronous operation. The state of the flip-flop changes immediately upon the application of the appropriate input level. No other conditions are necessary. This is not true of the J and K inputs. Their effect is dependent upon the state of the clock signal. Therefore, we call the J and K inputs synchronous because they cause state changes only on the occurrence of a specific clock transition, that is in synchronism with the clock.

When the clock input switches from 1 to 0, state changes occur. It is on this transition that the contents of the master latch is transferred to the slave latch. For some types of JK flip-flops, toggling occurs on the leading or positive edge of the clock signal. Be sure to check the manufacturer's data sheet for details on any device you are using.

The synchronous operation of the JK flip-flop is summed up in the truth table below. Note that only the normal (Q) output condition is shown, but it is given twice, once prior to a clock pulse (t) and then after one clock pulse (t + 1). The output state X can represent either set (1) or reset (0).

INPUTS		OUTPUTS	
J	K	Q(t)	Q(t + 1)
0	0	X	X
0	1	X	0
1	0	X	1
1	1	X	\bar{X}

Refer to the truth table. When the J and K inputs are both low, the clock can change all it wants to but it will not affect the state of the flip-flop. The flip-flop simply retains its previous condition which can be either set or reset. This is an inhibit mode.

To reset the JK flip-flop, apply a 0 to the J input and a 1 to the K input then apply a clock pulse. The flip-flop will reset. To set the JK flip-flop you apply a 1 to J and 0 to K and again apply a clock pulse. The flip-flop will set on the trailing edge of the clock.

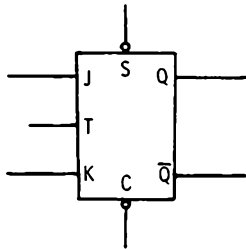


Figure 6-34

With the J and K inputs both at binary 1, the flip-flop toggles or complements each time the clock switches from 1 to 0. The flip-flop acts as a 2 to 1 frequency divider. Modern integrated circuit flip-flops are available in a variety of configurations. Some ECL flip-flops can toggle at rates as high as 1 GHz.

That completes the basic operation of a JK flip-flop. The symbol used to represent it is shown in Figure 6-34.

As a final check of your understanding of this important device, consider the input waveforms shown in Figure 6-35.

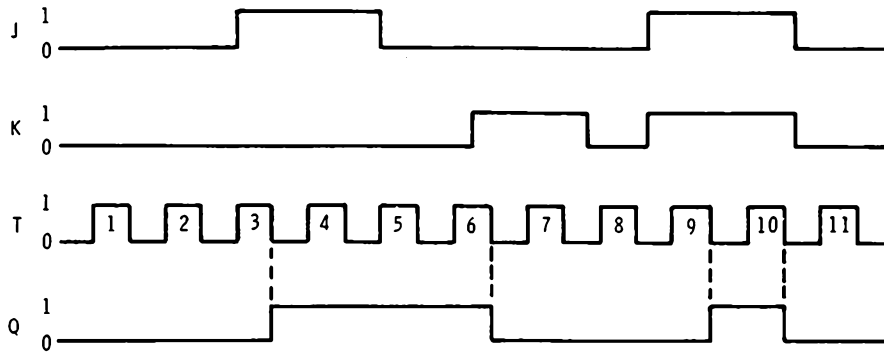


Figure 6-35

The J and K inputs affect the flip-flop state, but state changes occur only on the 1 to 0 transition of the clock pulse. This is synchronous operation.

In Figure 6-35, the normal output of the flip-flop is low prior to the occurrence of the first clock (T) pulse. Pulses 1 and 2 occur but since both the J and K inputs are low, the flip-flop is inhibited and no state change takes place. Then, the J input goes high. On the trailing edge of the next clock pulse (3), the flip-flop sets. When pulse 4 occurs, J is still high so the flip-flop would remain set. The J input goes low, then clock pulse 5 occurs. With both J and K low, the trailing edge of pulse 5 has no effect. The flip-flop remains set. Next, the K input goes high and J remains low. On the occurrence of the 1 to 0 transition of pulse 6, the flip-flop resets. Pulse 7 tries to reset it again. The K input goes low. Pulse 8 occurs and since J and K are low, the flip-flop remains reset. The J and K inputs go high simultaneously. Pulses 9 and 10 then toggle or complement the flip-flop, 9 setting it and 10 resetting it. After this J and K go low inhibiting the flip-flop. Pulse 11 has no effect.

The JK flip-flop is highly versatile. They are widely used in storage registers, shift registers, frequency dividers, and counters. You will learn more about each of these circuits in a later unit.

Self Test Review

17. The asynchronous inputs to a JK flip-flop are designated:
 - a. J and K
 - b. S and C
 - c. Q and \bar{Q}
 - d. T
18. The JK flip-flop operates as a NAND latch when which inputs are used?
 - a. S and C
 - b. J and K
 - c. T
 - d. none of the above
19. On a JK flip-flop the S and C inputs are high, the J input is high, the K input is low. What is the state of the flip-flop when one clock pulse occurs on the T input?
 - a. reset
 - b. set
 - c. ambiguous
 - d. insufficient information given to determine the state.
20. The state of the JK flip-flop changes when the clock signal on T switches from:
 - a. high to low
 - b. low to high
 - c. either a. or b.
21. The following conditions exist in a JK flip-flop: $J = K = 1$, $S = C = 1$, $\bar{Q} = 1$, $Q = 0$. What is the binary contents of the flip-flop after three clock pulses occur?
 - a. binary 0
 - b. binary 1
 - c. insufficient data given to determine state.
22. Both J and K inputs are held low. The S and C inputs are high. The Q output is 0. What is the state of the flip-flop after three clock pulses?
 - a. binary 0
 - b. binary 1
 - c. insufficient information given.

23. Which of the following conditions will reset a JK flip-flop? (indicate all choices that apply)
- a. $J = 1, K = 0, S = 1, C = 1, T$ changes
 - b. $J = 1, K = 1, S = 1, C = 1, T$ changes
 - c. $J = 0, K = 1, S = 1, C = 1, T$ changes
 - d. $J = 0, K = 0, S = 1, C = 0, T$ changes
 - e. $J = 1, K = 1, S = 0, C = 1, T$ changes
 - f. $J = 1, K = 0, S = 0, C = 0, T$ changes
24. Disregarding the S and C inputs, a JK flip-flop changes state when
- a. J changes
 - b. K changes
 - c. J and K change
 - d. when T switches from 1 to 0.
25. In a JK flip-flop, $J = K = 1, S = C = 1$. The T input is a 330 KHz square wave. The Q output is a
- a. binary 0
 - b. binary 1
 - c. 165 KHz square wave
 - d. 330 KHz square wave.
26. In a JK flip-flop, $J = K = 1, S = C = 1$. The T input is at 2 MHz with a duty cycle of 30 percent. What is the frequency and duty cycle of the Q output?
- a. 2 MHz, 30 percent
 - b. 2 MHz, 15 percent
 - c. 1 MHz, 15 percent
 - d. 1 MHz, 30 percent
 - e. 1 MHz, 50 percent.

NOTE: The duty cycle is the ratio of the pulse on (binary 1) time to the period of the signal times 100 percent

$$\text{percent duty cycle} = \frac{\text{pulse on time}}{\text{period}} \times 100 \text{ (period} = 1/f\text{)}$$

27. A JK flip-flop could be used for switch contact bounce buffering?
- a. True
 - b. False

Answers

- 17. b. S and C
- 18. a. S and C
- 19. b. set
- 20. c. Could be either. Check data sheet to be certain.
- 21. b. binary 1. With $\bar{Q} = 1$ and $Q = 0$, the flip-flop is initially reset. The first clock pulse toggles the flip-flop to a binary 1, the second to binary 0, and the third back to binary 1.
- 22. a. binary 0. With $J = K = \text{low}$ (binary 0) the flip-flop will not toggle.
- 23. c. $J = 0, K = 1, S = 1, C = 1$, T changes (synchronous)
d. $J = 0, K = 0, S = 1, C = 0$, T changes (asynchronous)
- 24. d. when T switches from 1 to 0
- 25. c. 165 KHz square wave. A JK flip-flop divides by 2.
- 26. e. 1 MHz 50 percent. The JK flip-flop divides by 2. The output always has a 50 percent duty cycle (equal binary 0 and binary 1 times) if the T input is a fixed frequency. Regardless of the duty cycle of the input signal, the flip-flop toggles on the 1 to 0 transition making the duration of the set and reset states equal to the period of the input.
- 27. a. True. Use the S and C inputs.

EXPERIMENT 9

Set-Reset Flip-Flops

OBJECTIVES: *To demonstrate the operation and characteristics of a set-reset (latch) flip-flop.*

Materials Needed:

Heathkit Digital Design Experimenter (ET-3200)
 1 — 74LS00 IC (443-728)
 1 — 74LS02 IC (443-779)

Procedure

1. Wire the latch circuit shown in Figure 6-36. The set (S) and reset (R) inputs to the latch will come from the \bar{A} and \bar{B} outputs of the two logic switches. The A logic switch is the set input, the B logic switch is the reset input. The latch outputs, C and \bar{C} , will be displayed on LED indicators L1 and L2 respectively. Be sure to apply power to the IC by connecting pin 14 to +5 volts and pin 7 to GND.

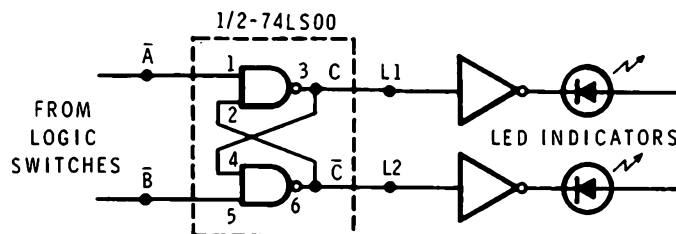


Figure 6-36

When the logic switches are not depressed, what is the normal state of the S and R inputs? _____.

2. Apply power to the Experimenter and note the state of the latch by observing LED indicator L1. L1 = binary _____.

Using the logic switches, apply the logic levels designated in Table I to the S and R inputs of the latch. Observe the output conditions on the LED indicators for each set of input states. Record your output states in Table I.

Table I

INPUTS		OUTPUTS		
S (A)	R(B)	C(L1)	\bar{C} (L2)	STATE
1	1			
1	0			
0	1			
0	0			

In the column marked STATE in Table I, write a single word designating the state represented by each set of outputs.

To get a feel for how the circuit operates, play with the inputs while observing the outputs. By repeatedly putting the latch into the set, reset, and ambiguous states you will understand it better.

3. Construct the circuit shown in Figure 6-37. Use a type 74LS02 IC. As before, the set and reset input signals will come from the logic pushbutton A (set) and B (reset).

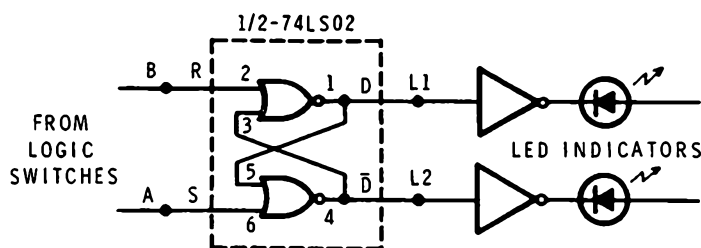


Figure 6-37

4. Apply power to the circuit. Note the state of the latch by observing L1. L1 = binary _____. Then apply the inputs given in Table II. Observe the outputs for each set of inputs and complete the D and \bar{D} columns.

Table II

INPUTS		OUTPUTS		
S (A)	R (B)	D	\bar{D}	STATE
0	0			
0	1			
1	0			
1	1			

In the column labeled STATE in Table II, write a word that designates the state of the latch as indicated by each of the output indications.

5. Compare the data in Tables I and II and note the similarities and differences in operation between the NAND and NOR latches.

Discussion

In Steps 1 and 2 you constructed a NAND gate latch using a type 74LS00 IC. The inputs were obtained from the momentary contact logic switches A and B. Specifically, the \bar{A} and \bar{B} outputs of these logic switches were used to supply the set and reset inputs respectively. The \bar{A} and \bar{B} outputs are normally high. When the switch is depressed, the \bar{A} or \bar{B} output goes low. With both switches in their normal or non-depressed state, the S and R inputs to the latch are binary 1. Therefore, they have no effect on the state of the latch.

When power is applied the latch can go into either state. The C output could be either 0 or 1 as indicated by LED indicator L1. Regardless of the initial state, C and \bar{C} should be complementary.

When the B logic switch is actuated, a low level is applied to the reset input. The C output (L1) should go low and the \bar{C} (L2) output high. When the B switch is released, the flip-flop will remain reset thereby storing a binary 0. Actuating the A logic switch supplies a low to the set input. The C output (L1) goes high and the \bar{C} output (L2) low. When the A switch is released, the flip-flop remains set storing a binary 1.

When both logic switches are actuated to apply a low to both set and reset inputs, both C and \bar{C} outputs go high. This is the ambiguous state.

In Steps 3 and 4 you constructed and tested a NOR gate latch. The set and reset inputs are supplied by the normal outputs (A and B) of the two logic switches on your Experimenter. The A and B outputs are normally low when the switches are not actuated. A low input or NOR latch does not change its state as it does in the NAND latch. Therefore, with both inputs low, the latch can be either set or reset depending upon the arbitrary state it comes up in when power is applied.

When the B logic switch is actuated, the B output goes high applying a high or binary 1 level to the reset input. This forces the D output low and the \bar{D} output high thereby indicating that a binary 0 has been stored. Upon releasing the B switch, the latch retains the reset state.

When you depress the A logic switch, you apply a high level to the set input. The D output goes high and the \bar{D} output goes low. The flip-flop remains in the set state when the switch is released.

If you apply binary 1's to both set and reset inputs at the same time by simultaneously actuating the A and B logic switches, the latch goes into the ambiguous state. Both outputs go low.

In comparing the NAND and NOR gate latches we can say in summary:

1. Either type flip-flop will store one bit of data, with the state of the outputs indicating the value of the bit stored. The two outputs are complementary.
2. The NAND latch requires a low level on either input to set or reset it. The NOR latch requires a high level at the appropriate input to change its state.
3. For a NAND latch, the inputs normally reset in the high state. The NOR latch inputs are normally both low.
4. Both types of latches have an ambiguous state. In the NAND both outputs are high. In the NOR gate latch, both outputs are low.

EXPERIMENT 10

D Type Flip-Flops

OBJECTIVES: *To demonstrate the operation of a D type flip-flop and a storage register*

Materials Needed:

Heathkit Digital Design Experimenter ET-3200

1 — 74LS00 TTL IC (443-728)

1 — 4001 CMOS IC (443-695)

1 — 74LS75 TTL IC (443-781)

Procedure

1. Wire the circuit shown in Figure 6-38. Use a type 74LS00 IC. Because of the large number of connections required, take your time to avoid making a wiring mistake. Double check your connections before you perform the experiment. Don't forget to connect pin 14 to +5 volts and pin 7 to GND. The D and T inputs will be supplied by data switches SW1 and SW2. The flip-flop outputs will be monitored on LED indicators L1 and L2.

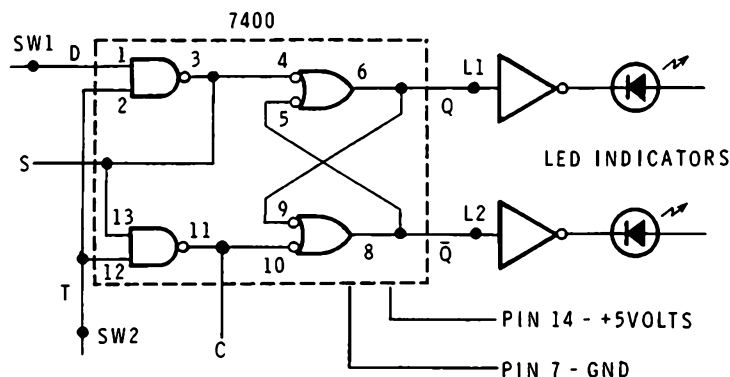


Figure 6-38

- Set input data switches to binary 0 and record the state of the flip-flop. Apply the logic levels indicated in Table I to the D and T inputs. Note the output states for each set of inputs and record your results in Table I.

TABLE I

INPUTS		OUTPUTS	
D	T	Q	\bar{Q}
0	0		
0	1		
1	0		
1	1		

- For a more graphic indication of exactly what takes place, replace the data switch on the D input with a logic clock signal. Set the clock frequency to 1 Hz and connect one of the free LED logic indicators to monitor it. Set the T input first to binary 1 and observe the flip-flop outputs for a brief period. Note the relationship between the clock state and the Q output. Set the T input to binary 0 and again observe the outputs. Repeat.
- Construct the circuit shown in Figure 6-39. Use a 4001 CMOS NOR IC. Use data switches for the D and T inputs and LED indicators for the outputs.

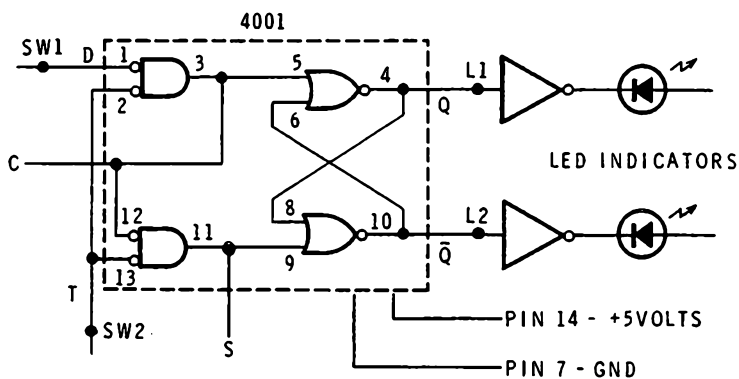


Figure 6-39

- Apply the logic levels given in Table II to the D and T inputs. Note the output states for each set of inputs and record your data in Table II.

TABLE II

INPUTS		OUTPUTS	
D	T	Q	\bar{Q}
0	0		
0	1		
1	0		
1	1		

6. Repeat Step 3 for the NOR D-type flip-flop.
7. Compare your results in Tables I and II and in Steps 3 and 6.
8. Wire the circuit shown in Figure 6-40. The 74LS75 IC contains four TTL D-type flip-flops similar in operation to the NAND D-type flip-flops

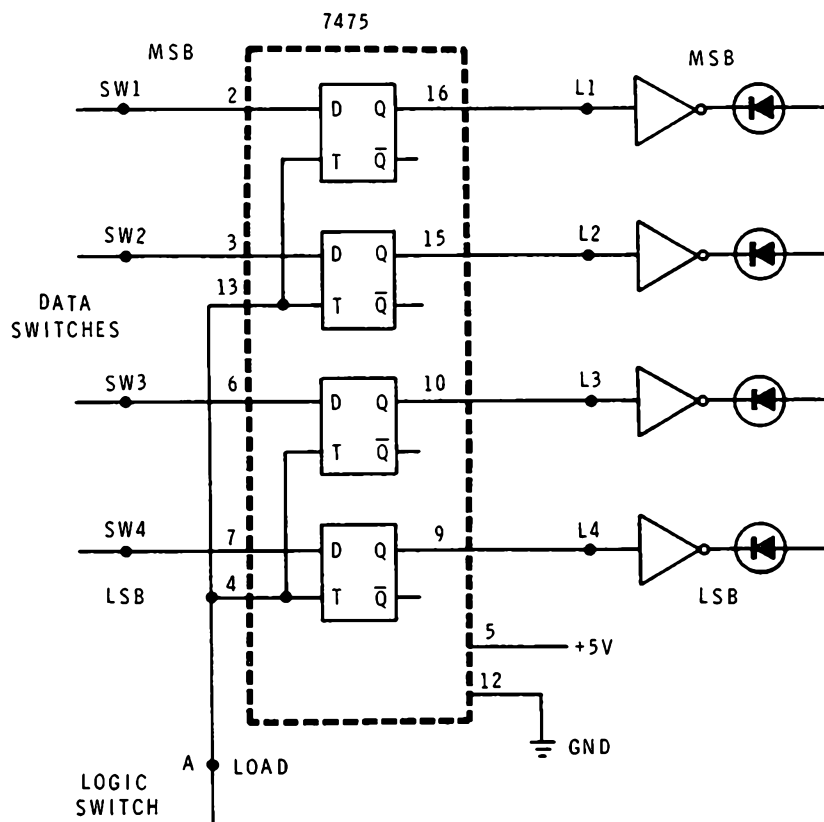


Figure 6-40

discussed earlier. Figure 6-41 shows the internal structure and pin connections for this device. Note that +5 volts is connected to pin 5 and GND is connected to pin 12. The data switches on the Experimenter will be used as a switch register. The switch outputs will be used as a source of data for a four bit register made from the flip-flops in the 74LS75. You will monitor the register output on the LED logic indicator.

Logic switch A will be used as the LOAD or strobe signal which transfers input data into the register.

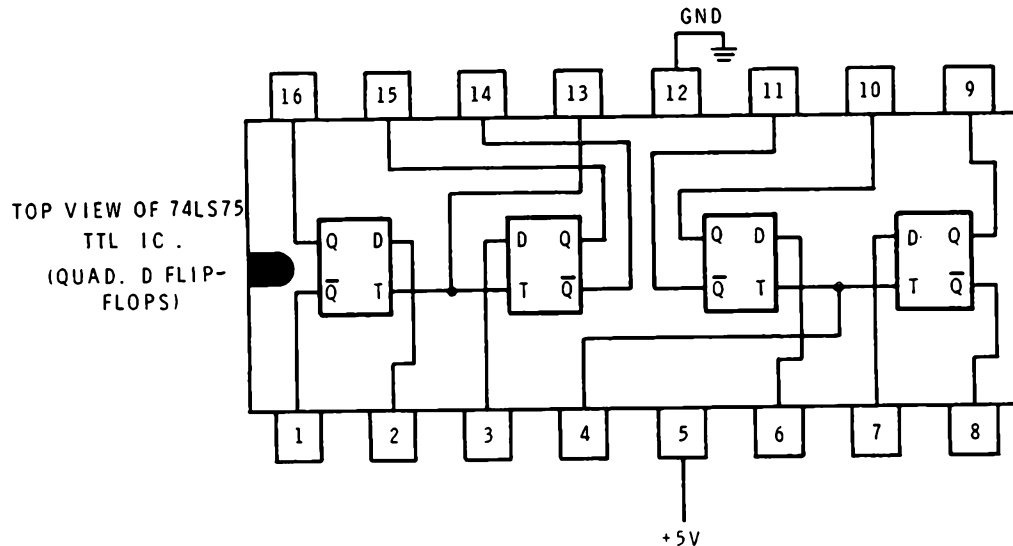


Figure 6-41
Top view of 74LS75 TTL IC.
(quad D flip-flops).

9. Apply power to the circuit and record the number in the register. Indicator L4 monitors the LSB. _____
10. Set all of the data switches to binary 0. Then momentarily depress the A logic switch. Record the binary number in the register. _____
11. Set all the data switches to binary 1. Depress the A logic switch and note the register contents. Record. _____
12. Load the sixteen binary numbers 0000 through 1111 into the register one at a time by setting the data switches then actuating the A logic switch. Verify that the input does load by comparing the LED indicator states with the data switch setting *after* the A button is depressed.

Discussion

In Step 1 you constructed a D-type flip-flop with TTL NAND gates. When you applied power in Step 2, the flip-flop could have assumed either the set or reset state. When you set the T input to binary 1 and the D input to binary 0, the flip-flop will be reset as indicated by L1 being off. With both the D and T inputs binary 1, flip-flop will become set. L1 will be on. With the T input set to binary 1, you can switch the D input between binary 0 and 1 and watch the normal output follow it. If the T input is binary 0, the flip-flop state will be that determined by previous inputs. Switching the D input between 0 and 1 while T is binary 0 will not affect the state of the flip-flop.

In Step 3 you applied the 1 Hz clock signal to the D input and observed the operation of the flip-flop. With T set to binary 0, the clock signal at the D input is ignored. But with T set to binary 1, the flip-flop output follows the D input. The LED indicators on the CLK and Q lines should switch off and on in synchronism.

In Step 4 you assembled a D flip-flop from CMOS NOR gates. In Steps 5 and 6 you evaluated its operation. Basically you should have found that its operation was identical to that of the NAND D flip-flop with the exception of the state of the T input. On the NOR flip-flop, the T input must be low in order for the D input to be recognized. With the T input low, the normal output follows or tracks the D input. When the T input is high, the D input will have no effect on the state of the flip-flop.

In Step 8 you assembled a 4-bit storage register using the four D-type flip-flops in a type 74LS75 TTL MSI IC. The operation of the flip-flops in this device is similar to the NAND D flip-flop you studied earlier. The data switches on the Experimenter were used as a switch register. The A logic switch is used as a manual LOAD control. The A output is normally low thereby keeping the T inputs to all four register flip-flops low. The inputs from the switch register are ignored. When the A switch is actuated, the A output goes high causing the data from the switch register to be loaded into the register.

When you first applied power, the contents of the register could have been anything. When power is applied to a flip-flop it can come up in either the set or reset condition. Next you reset the register by loading all binary 0's. Then you loaded 1111. These two operations check to see that all four flip-flops work in both states.

Finally, you sequentially loaded the numbers 0000 through 1111. This gives you an opportunity to become familiar with setting binary numbers on the switch registers and practice in reading binary numbers from the LED indicators. An important point you should have grasped is that the input word can be different from the register contents. With the LOAD input low, the D inputs to the flip-flop are ignored. When LOAD is made binary 1, however, the register output becomes equal to the inputs.

EXPERIMENT 11

JK Flip-Flops

OBJECTIVES:

To demonstrate the operation and characteristics of a JK flip-flop.

Materials Needed:

Heathkit Digital Design Experimenter

1—74LS04 TTL IC (443-755)

1—74LS76 TTL IC (443-829)

Procedure

1. Connect the circuit shown in Figure 6-42. Use data switches for the J, K, S, and C inputs. Use logic switch A for the clock T input. Connect LED indicators to each output. The pin connections for 74LS76 dual JK flip-flop are

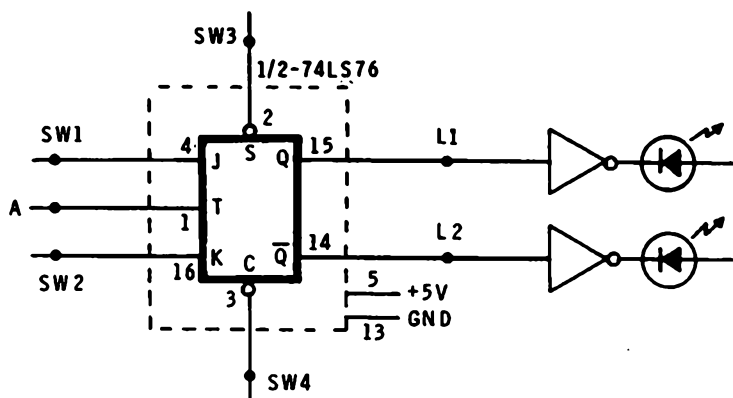


Figure 6-42

shown in Figure 6-43. There are two identical JK flip-flops in the 74LS76 IC, but we will use only one. Connect + 5 volts to pin 5 and GND to pin 13.

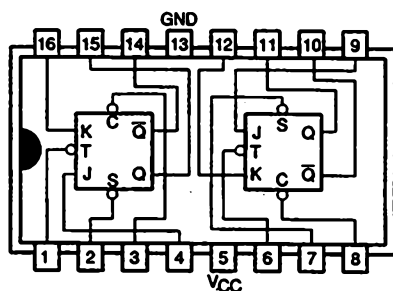


Figure 6-43

Pin connections for dual JK flip-flop 74LS76.

2. First you will check the asynchronous operation of the JK flip-flop. Set $J = K = 1$ with SW1 and SW2. Apply the levels indicated in Table I to the S and C inputs. Note the output states and record them in Table I. Repeat this step with $J = K = 0$. Record the results in Table I.

Do the JK inputs affect the asynchronous operation? _____

TABLE I

		J = K = 1		J = K = 0	
INPUTS		OUTPUTS		OUTPUTS	
S	C	Q	\bar{Q}	Q	\bar{Q}
1	1				
0	1				
1	0				
0	0				

3. Next, verify the synchronous operation of the JK flip-flop. Remove the wire connecting the \bar{Q} output to LED indicator L2. Set the S and C inputs to binary 1. Then apply the logic levels indicated in Table II. Note the normal output state before (Q) and after [$Q(t+1)$] the application of a single clock pulse from the A logic switch. After you have completed Table II, repeat the inputs given and toggle the clock (T) input several times with the A logic switch for each set of inputs. Note the results on the LED logic indicators.

NOTE: $Q(t+1)$ means the state of the Q output after the application of one clock pulse with the given inputs.

TABLE II

INPUTS		OUTPUTS	
J	K	Q	Q(t+1)
0	0		
0	1		
1	0		
1	1		

4. Set the J and K inputs to binary 1, and the S and C inputs to binary 1 with the logic switches. Remove the A logic switch from the T input and connect a 1 Hz clock (CLK) signal to it. Also connect a spare LED logic indicator to monitor the CLK signal. Observe the CLK input and Q output on the LED indicators. What is the relationship between input and output frequencies? _____
5. Construct the circuit shown in Figure 6-44. The circuit will be driven from the 1 Hz CLK signal. The A and B logic switches will control the circuit. You will observe the output states on LED indicators L3 and L4, and the CLK input on LED indicator L1.

NOTE: The logic level of the JK flip-flop's Q output is not strong enough to drive both the next flip-flop and an LED. Therefore, you will use a 74LS04 hex inverter on the \bar{Q} output to drive the LED.

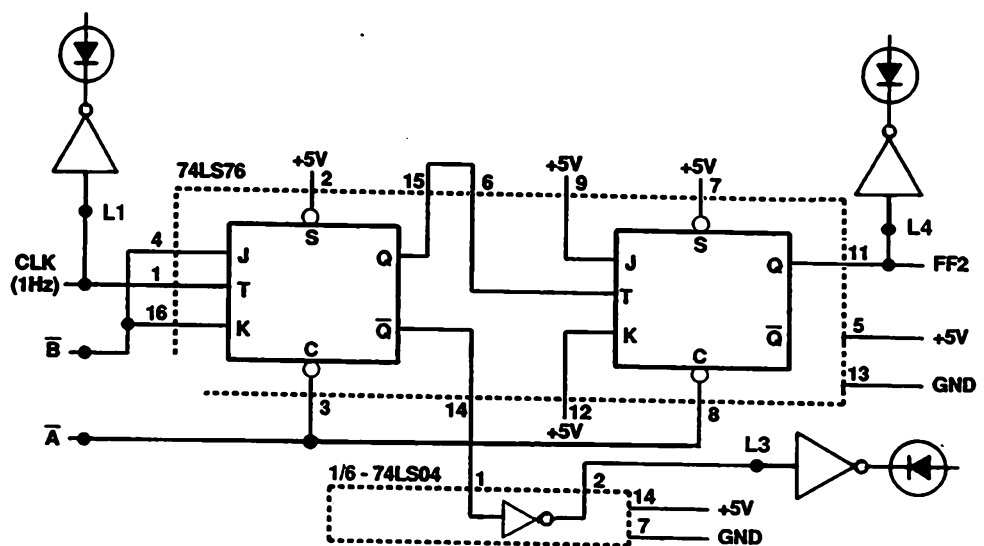


Figure 6-44

6. Observe the relationship between the input (L1) and output (L3 and L4) waveforms. You can do this by counting the number of input and output pulses. Sketch a timing diagram illustrating this relationship.

7. While the circuit is operating, depress and hold the B logic switch. What effect does this have on the circuit? (Note the output states). Release the B switch. Repeat this step several times.
8. Depress and hold the A logic switch while the circuit is operating. Note the effect on the outputs. Release the A switch. Repeat several times.

NOTE: If you have an oscilloscope, set the CLK frequency to 1 KHz or 100 KHz and observe CLK, FF1, and FF2, noting their frequency relationship.

Discussion

In Steps 1 and 2, you verified the operation of the JK flip-flop in the asynchronous mode. This refers to the use of the set (S) and clear (C) or reset inputs to control the state of the flip-flop. From the data you recorded in Table I, you should have found that the JK flip-flop functions just like a NAND latch when the S and C inputs are used. With both inputs binary 1, the flip-flop can be in either state. When C is low and S is high, the flip-flop is reset. With C high and S low, the flip-flop is set. If both S and C are low, the ambiguous state ($Q = \bar{Q} = 1$) occurs.

In steps 3 and 4, you verified the synchronous operation of the JK flip-flop. The most important points to note are:

1. It is not the JK inputs that cause the state of the flip-flop to change. It is the T input 1 to 0 transition that causes the state change. The J and K inputs do determine the state to which the flip-flop goes but not *when* it changes.
2. The flip-flop toggles or complements each time a 1 to 0 change occurs on the T input with $J = K = 1$.
3. The flip-flop does not toggle when a clock pulse occurs if $J = K = 0$. This makes the JK inputs useful as a toggle inhibit control.
4. To reset the JK flip-flop, apply a 0 to the J input and a 1 to the K input, then apply a clock pulse.
5. To set the JK flip-flop, apply a 1 to the J input and a 0 to the K input, then apply a clock pulse.
6. For every two binary 1 input pulses on T, one binary 1 pulse occurs at the Q output. This indicates a two-to-one frequency division.

In Step 5 you cascaded two JK flip-flops and in Step 6 you determined the input-output relationships. By observing the LED indicators you should have found that for every four binary 1 clock input pulses there were two pulses from FF1 and one pulse from FF2. This indicates that each flip-flop divides the input frequencies by 2. The overall circuit, both flip-flops together, divides by 4. The input is 1 Hz. The output of FF1 is 0.5 Hz and the output of FF2 is 0.25 Hz. Your input-output waveforms should appear as shown in Figure 6-45.

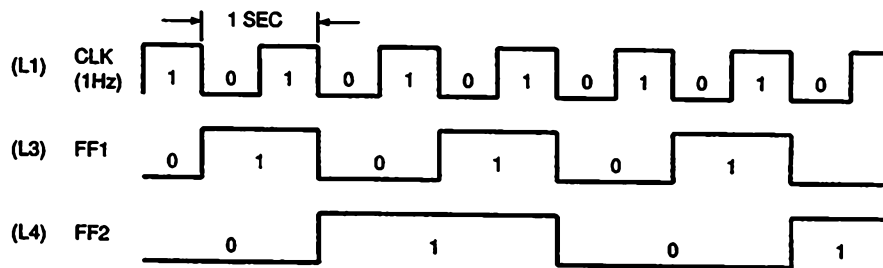


Figure 6-45

In Step 7 you used the B logic switch to control the JK inputs to FF1. With the switch in its normal position, the \overline{B} output is binary 1. Therefore, the JK inputs are binary 1 and the flip-flop toggles with each clock pulse. When you depress the B logic switch, \overline{B} goes low. This inhibits FF1. The clock pulses will not affect it with $J = K = 0$. It will simply retain the state to which it was set by the last clock pulse prior to the JK inputs becoming low. Since FF1 does not toggle, FF2 will not toggle. The input to FF2 comes from FF1. As a result, when the JK inputs on FF1 go low, the clock pulse has no effect and the flip-flop states can be anything

When you depress the A logic switch you reset both flip-flops. With the A logic switch in its normal position, the \overline{A} output is high. This puts a binary 1 on both C inputs. This will not affect the flip-flop states. When you depress the A logic switch, the \overline{A} output goes low. This resets both flip-flops immediately. Regardless of the states of the flip-flops, when you depress A, both will be put into the binary 0 condition. You will note that this reset state overrides the clock signal. With the C inputs low, the clock input has no effect. In a JK flip-flop, the asynchronous inputs always take precedence over the synchronous inputs.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and answer all the questions first before checking the answers.

1. A flip-flop is a logic element that:
 - a. makes decisions
 - b. stores binary data
 - c. generates a clock signal
 - d. buffers NAND gates
2. Which of the following is **not** a type of flip-flop?
 - a. RS
 - b. one shot
 - c. latch
 - d. D
 - e. register
 - f. JK
3. (A) The normal output of a flip-flop is low. What state is the flip-flop in?
 - a. set
 - b. reset(B) The complement output of a flip-flop is binary 0. What state is the flip-flop in?
 - a. set
 - b. reset
4. A JK flip-flop toggles:
 - a. On the positive or leading edge of the clock pulse.
 - b. When the clock switches from binary 0 to binary 1.
 - c. When the J and K inputs simultaneously switch from binary 1 to binary 0.
 - d. On the negative or trailing edge of the clock pulse.
5. A storage register made up of six D flip-flops is storing a binary word. The flip-flop states are: A = set, B = set, C = reset, D = set, E = reset, F = set. The A flip-flop is the LSB. The decimal equivalent of the register content is:
 - a. 43
 - b. 47
 - c. 54
 - d. 59

6. Write the name of the flip-flop corresponding to each symbol in Figure 6-46.

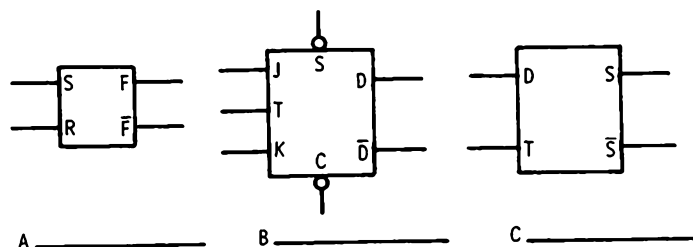


Figure 6-46

Illustration for Question 6.

7. The "limbo" state of a flip-flop is recognizable when:
- both outputs are binary 0.
 - both outputs are binary 1.
 - both outputs are the same.
 - the outputs are complementary.
8. A digital circuit used to store a binary number is called a:
- flip-flop
 - memory
 - word
 - register
9. The name given to the class of logic circuits containing flip-flops is:
- combinational
 - sequential
 - linear
 - feedback
10. To reset a positive NOR latch which of the following conditions must occur?
- binary 0 to S input
 - binary 1 to S input
 - binary 0 to R input
 - binary 1 to R input

11. To preset a JK flip-flop to the binary 1 state, which of the following conditions must occur?
 - a. $J = 1, K = 0$
 - b. $S = 0, C = 1$
 - c. $J = 0, K = 1$
 - d. $S = 1, C = 0$
12. Which of the following ways can a JK flip-flop be reset? Check all that apply.
 - a. Ground the C input.
 - b. Ground the S input.
 - c. Set J to 0 and K to 1 and apply a clock pulse.
 - d. Set J to 1 and K to 0 and apply a clock pulse.
 - e. Toggle the T input with $J = K = 1$.
13. One common application of the RS flip-flop is:
 - a. switch contact debouncing
 - b. registers
 - c. frequency division
 - d. counting
14. To inhibit a JK flip-flop from toggling, which of the following must occur?
 - a. $J = 0, K = 1$
 - b. $J = 1, K = 0$
 - c. $J = K = 0$
 - d. $J = K = 1$
15. D flip-flops are most frequently used in:
 - a. switch contact debouncing
 - b. storage registers
 - c. frequency dividing
 - d. counting
16. The normal output of a JK flip-flop is 48 MHz. The clock input is:
 - a. 24 MHz
 - b. 48 MHz
 - c. 71 MHz
 - d. 96 MHz
17. The term used to designate a logic circuit that is actuated by a clock signal is:
 - a. sequential
 - b. synchronous
 - c. pulsed
 - d. asynchronous

18. Complete the Q (normal) output waveform of an RS and NAND flip-flop, given the S and R inputs shown in Figure 6-47.

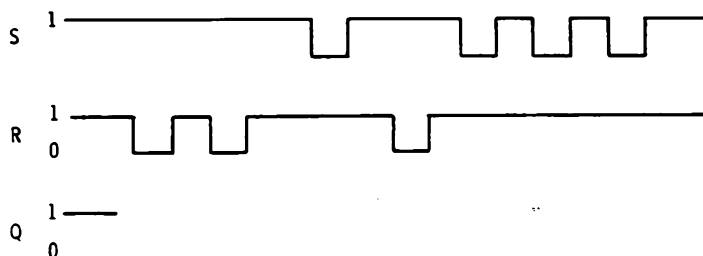


Figure 6-47

Illustration for Question 18.

19. The Q output of the D flip-flop shown in Figure 6-48 is correct for the given inputs.
- True
 - False

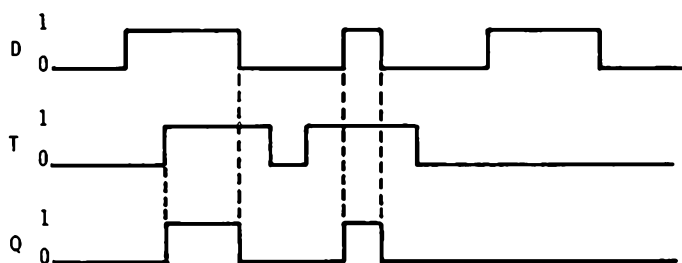


Figure 6-48

Illustration for Question 19.

20. Given the J, K, and T inputs to a JK flip-flop shown in Figure 6-49, draw the Q output waveform

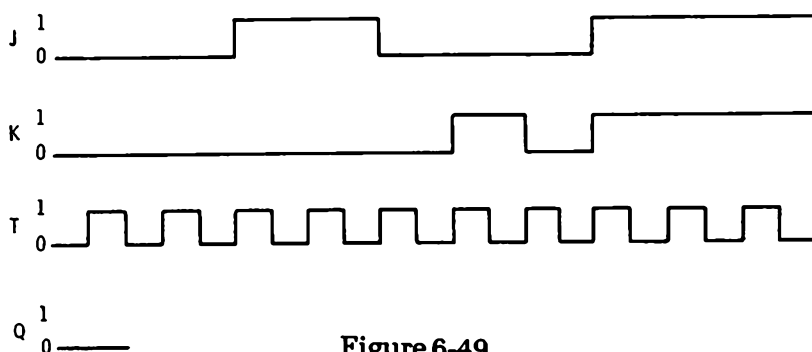


Figure 6-49

Illustration for Question 20.

EXAMINATION ANSWERS

1. b. stores binary data — on bit
2. e. register. A register is NOT a flip-flop.
3. (A) b. reset
(B) a. set
4. a,d Depends on the manufacturers data sheet.
5. a. 43 Register content FEDCBA = $101011_2 = 43_{10}$
6. A latch or RS
B JK
C D
7. c. Both outputs are the same: Binary 1 in a NAND latch, Binary 0 in a NOR latch.
8. d. register
9. b. sequential
10. d. binary 1 to R input
11. b. $S = 0, C = 1$
12. a. Ground the C input.
c. Set $J = 0$ and $K = 1$ and apply a clock pulse.
e. Toggle the T input with $J = K = 1$
13. a. switch contact debouncing
14. c. $J = K = 0$
15. b. storage registers
16. d. 96 MHz. The input is double the output, or the output is one-half the input.
17. b. synchronous

18. See Figure 6-50

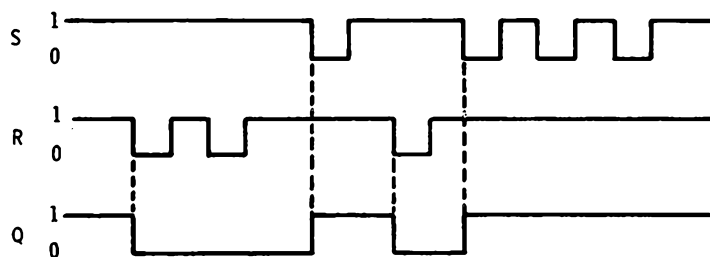


Figure 6-50
Solution to Question 18.

19. a. True

20. See Figure 6-51.

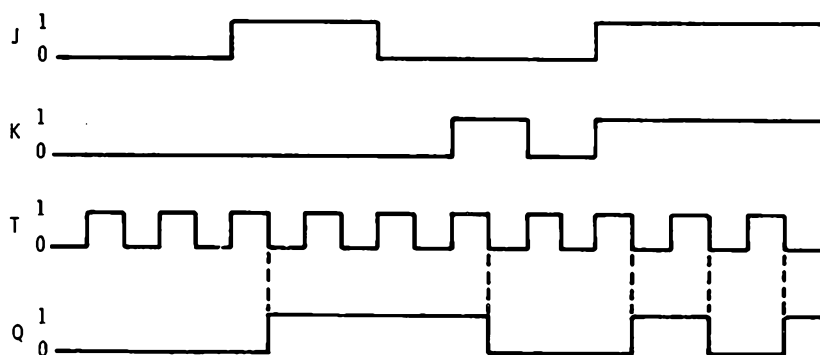


Figure 6-51
Solution to Question 20.

Unit 7

**SEQUENTIAL LOGIC CIRCUITS:
COUNTERS, SHIFT REGISTERS
AND CLOCKS**

CONTENTS

Introduction	7-3
Unit Objectives	7-4
Unit Activity Guide	7-5
Counters	7-6
BCD Counters	7-26
Special Counters	7-34
Shift Registers	7-43
Shift Register Applications	7-53
MOS Shift Registers	7-63
Clocks and One Shots	7-71
Experiment 12 — Binary Counters	7-88
Experiment 13 — The BCD Counter	7-97
Experiment 14 — Counter Applications	7-100
Experiment 15 — Shift Registers	7-105
Experiment 16 — Shift Register Applications	7-113
Experiment 17 — Clocks and One Shots	7-123
Unit Examination	7-131
Examination Answers	7-135

INTRODUCTION

In this unit you are going to learn about sequential logic circuits. These are logic circuits that are used for a variety of timing, sequencing, and storage functions. The key characteristic of sequential logic circuits is memory. Sequential logic circuits are capable of storing binary data. The output of a sequential logic circuit is a function not only of the various input states applied to the circuit but also the result of previous operations which are stored in the circuit itself.

The main circuit elements of a sequential circuit are flip-flops. These flip-flops store binary data and their states are changed by the logic input signals in accordance with the current information stored in them. The sequential logic operations are generally sequenced by a periodic logic signal known as a clock. The clock is an oscillator that generates rectangular pulses at a fixed frequency.

As with combinational logic circuits, there can exist almost an infinite variety of sequential logic circuits. However, in practice only a few special types seem to regularly reoccur. The two most commonly used sequential circuits are counters and shift registers. Because these two types of sequential circuits are the most widely used, we will emphasize their operation and application in this unit. In addition, clock circuits and special sequential circuit components such as one shot multivibrators and the delay lines will also be considered. Like combinational logic circuits, most sequential circuits are implemented with integrated circuits. In fact, most of the commonly used sequential circuits are available as a single ready to use MSI logic package. We will concentrate on these popular devices in this unit.

UNIT OBJECTIVES

When you complete this unit you will be able to:

1. Name the two most widely used types of sequential logic circuits.
2. Explain the operation of both binary and BCD counters.
3. Determine the maximum count capability of a binary or BCD counter given the number of flip-flops.
4. Determine the count sequence of a counter from a logic diagram and draw the circuit waveforms.
5. Explain the operation of a shift register.
6. List four applications for shift registers.
7. Explain the purpose of the clock signal and show a method of developing it.
8. Explain the operation of a one shot and list several applications.

UNIT ACTIVITY GUIDE

Completion
Time

- | | |
|---|-------|
| <input type="checkbox"/> Read "Counters." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 1–15. | _____ |
| <input type="checkbox"/> Read "BCD Counters." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 16–24. | _____ |
| <input type="checkbox"/> Read "Special Counters." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 25 and 26. | _____ |
| <input type="checkbox"/> Read "Shift Registers." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 27–32. | _____ |
| <input type="checkbox"/> Read "Shift Register Applications." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 33–38. | _____ |
| <input type="checkbox"/> Read "MOS Shift Registers." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 39–44. | _____ |
| <input type="checkbox"/> Read "Clocks and One Shots." | _____ |
| <input type="checkbox"/> Answer Self Test Review Questions 45–50. | _____ |
| <input type="checkbox"/> Perform Experiment 12. | _____ |
| <input type="checkbox"/> Perform Experiment 13. | _____ |
| <input type="checkbox"/> Perform Experiment 14. | _____ |
| <input type="checkbox"/> Perform Experiment 15. | _____ |
| <input type="checkbox"/> Perform Experiment 16. | _____ |
| <input type="checkbox"/> Perform Experiment 17. | _____ |
| <input type="checkbox"/> Complete the Unit Examination. | _____ |
| <input type="checkbox"/> Review the Examination Answers. | _____ |

COUNTERS

A binary counter is a sequential logic circuit made up of flip-flops that are used to count the number of binary pulses applied to it. The pulses or logic level transitions to be counted are applied to the counter input. These pulses cause the flip-flops in the counter to change state in such a way that the binary number stored in the flip-flops is representative of the number of input pulses that have occurred. By observing the flip-flop outputs you can determine how many pulses were applied to the input.

There are several different types of counters used in digital circuits. The most commonly used is the binary counter. This type of counter counts in the standard pure binary code. BCD counters which count in the standard 8421 BCD code are also widely used. In addition, counters can be developed to count in any of the special binary or BCD codes in common use. Both up and down counters are available.

Binary Counters

A binary counter is a sequential logic circuit that uses the standard pure binary code. Such a counter is made up by cascading JK flip-flops as shown in Figure 7-1. The normal output of one flip-flop is connected to the toggle (T) input of the next flip-flop. The JK inputs on each flip-flop are open or high. The input pulses to be counted are applied to the toggle input of the A flip-flop.

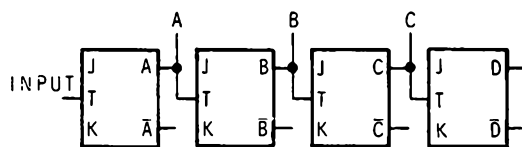


Figure 7-1
Four-bit binary counter.

To see how this binary counter operates, assume that a JK flip-flop toggles or changes state each time a trailing edge transition occurs on its T input. The flip-flop will change state when the normal output of the previous flip-flop switches from binary 1 to binary 0. If we assume that the counter is initially reset, the normal outputs of all the

flip-flops will be binary 0. When the first input pulse occurs, the A flip-flop will become set. The binary number stored in the flip-flops indicates the number of input pulses that have occurred. To read the number stored in the counter you simply observe the normal outputs of the flip-flops. The A flip-flop is the least significant bit of the word. Therefore, the four bit number stored in the counter is designated DCBA. After the first input pulse, the counter state is 0001. This indicates that one input pulse has occurred.

When the second input pulse occurs the A flip-flop toggles and this time becomes reset. As it resets its normal output switches from binary 1 to binary 0. This causes the B flip-flop to become set. Observing the new output state, you see that it is 0010 or the binary equivalent of the decimal number 2. Two input pulses have occurred.

When the third input pulse occurs the A flip-flop will again set. The normal output switches from binary 0 to binary 1. This transition is ignored by the T input of the B flip-flop. The number stored in the counter at this time is 0011 or the number 3 indicating that three input pulses have occurred.

When the fourth input pulse occurs, the A flip-flop is reset. Its normal output switches from binary 1 to binary 0 thereby toggling the B flip-flop. This causes the B flip-flop to reset. As it does, its normal output switches from binary 1 to binary 0 causing the C flip-flop to become set. The number now in the counter is 0100 or a decimal 4. This process continues as the input pulses occur. The count sequence is the standard 4 bit binary code as indicated in Figure 7-2.

An important point to consider is the action of the circuit when the number stored in the counter is 1111. This is the maximum value of a four bit number and the maximum count capacity of the circuit. When the next input pulse is applied, all flip-flops will change state. As the A flip-flop resets, the B flip-flop resets. As the B flip-flop resets it, in turn, resets the C flip-flop. As the C flip-flop resets, it toggles the D flip-flop which is also reset to zero. The result is that the contents of the counter becomes 0000. As you can see from Figure 7-2, when the maximum content of the counter is reached it simply recycles and starts its count again.

D	C	B	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

RECYCLE

Figure 7-2
Count sequence of four
bit binary counter.

The complete operation of the four bit binary counter is illustrated by the input and output waveforms in Figure 7-3. The upper waveform is a series of input pulses to be counted. Here they are shown as a periodic binary waveform but of course it is not necessary for the input signal to be of a constant frequency or have equally spaced input pulses. The waveforms also show the normal output of each flip-flop.

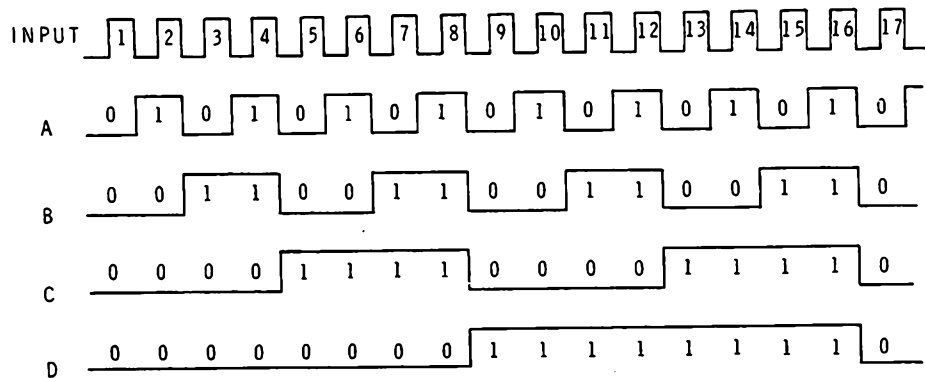


Figure 7-3

Input and output waveforms
of a four bit binary counter.

In observing the waveforms in Figure 7-3, you should note several important things. First, all the flip-flops toggle (change state) on the trailing edge or the binary 1 to binary 0 transition of the previous flip-flop. With this in mind you can readily trace the output of the first (A) flip-flop by simply observing when the trailing edges of the input occur.

The output of the B flip-flop is a function of its input which is the output of the A flip-flop. Note that its state change occurs on the trailing edge of the A output. The same is true of the C and D flip-flops. The binary code after each input pulse is indicated on the waveforms. Of course, this corresponds to the binary count sequence in Figure 7-2.

Frequency Divider. Another important fact that is clear from the waveforms in Figure 7-3 is that the binary counter is also a frequency divider. The output of each flip-flop is one half the frequency of its input. If the input is a 100 kHz square wave, the outputs of the flip-flops are:

- A – 50 kHz
- B – 25 kHz
- C – 12.5 kHz
- D – 6.25 kHz

The output of a pure binary counter is always some sub-multiple of two. The four bit counter divides the input by 16, ($100\text{kHz} \div 16 = 6.25\text{ kHz}$).

Maximum Count. The maximum count capability of a binary counter is a function of the number of flip-flops in the counter. The maximum number that can be contained in a binary counter before it recycles is determined in the same way that we can determine the maximum binary number that can be represented by a word with a specific number of bits. The formula below expresses the relationship between the number of flip-flops in a counter and its maximum count capability.

$$N = 2^n - 1$$

Here N is the maximum number that occurs prior to the counter recycling. The number of flip-flops is designated by n . For example, the maximum number that can be contained in a counter using four flip-flops is

$$N = 2^4 - 1 = 16 - 1 = 15 \text{ (binary 1111)}$$

Another example is a binary counter with nine flip-flops. A maximum count capability here then is

$$N = 2^9 - 1 = 512 - 1 = 511 \text{ (binary 111111111)}$$

To determine the number of flip-flops required to implement a counter with a known or required count capability, use the formula given below.

$$n = 3.32 \log_{10} N$$

For example, if you wish to implement a binary counter capable of counting to 100, you could determine the number of flip-flops as follows:

$$n = 3.32 \log_{10} 100 = 3.32 (2) = 6.64$$

$$n = 7$$

Since there is no such thing as a fractional part of a flip-flop, the next higher whole number value is used. A counter with 7 flip-flops has a maximum count capability of

$$2^7 - 1 = 127.$$

When the binary counter is used as a frequency divider, the factor by which the counter divides is a function of the number of flip-flops used. To determine the divide ratio, the expression below is used.

$$N = 2^n$$

For example, if the counter contained six flip-flops it would divide an input signal by

$$N = 2^6 = 64$$

What this means is that the output of the sixth flip-flop will be $\frac{1}{64}$ the frequency of the input signal applied to the first flip-flop. With a binary counter, the frequency division ratio is always some power of 2. As you will see later, it is possible to implement frequency dividers that can divide the frequency by any integer value.

Down Counters. The binary counter just described is referred to as an up-counter. Each time that an input pulse occurs, the binary number in the counter is increased by one. We say that the input pulses increment the counter. It is also possible to produce a down counter where the input pulses cause the binary number in the counter to decrease by one. The input pulses are said to decrement the counter.

D	C	B	A
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

Figure 7-5

Count sequence for four bit down counter.

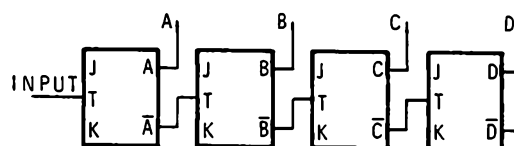


Figure 7-4

Four bit binary down counter.

A four bit binary down counter is shown in Figure 7-4. It is practically identical to the up-counter described earlier. The only difference is that the complement output rather than the normal output of each flip-flop is connected to the toggle input of the next flip-flop in sequence. This causes the count sequence to be the exact reverse of the up counter. The count sequence is illustrated in Figure 7-5. The waveforms associated with this counter are shown in Figure 7-6.

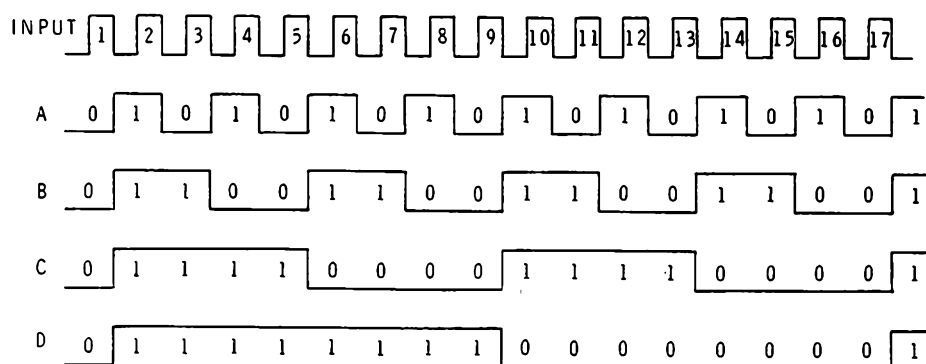


Figure 7-6
Input and output waveforms
of a four bit binary down counter.

In analyzing the operation of this counter, keep in mind that we still determine the contents of the counter by observing the normal outputs of the flip-flops as we did with the up-counter. Assuming the counter is initially reset and its contents are 0000, the application of an input pulse will cause all flip-flops to become set. With the A flip-flop reset, its complement output is high. When the first input pulse is applied, the A flip-flop will set. As it does its complement output will switch from binary 1 to binary 0 thereby toggling the B flip-flop. The B flip-flop becomes set and its complement output also switches from binary 1 to binary 0. This causes the C flip-flop to set. In the same way the complement output of the C flip-flop switches from high to low thereby setting the D flip-flop. The counter recycles from 0000 to 1111.

When the next input pulse arrives, the A flip-flop will again be complemented. It will reset. As it resets the complement output will switch from binary 0 to binary 1. The B flip-flop ignores this transition. No further state changes take place and the content of the counter is 1110. As you can see this input pulse causes the counter to be decremented from 15 to 14.

Applying another input pulse again complements the A flip-flop. It now sets. As it sets, its complement output switches from binary 1 to binary 0. This causes the B flip-flop to reset. As it resets, the complement output switches from binary 0 to binary 1. The C flip-flop ignores this transition. The counter now contains 1101 or 13. Again the input pulse caused the counter to be decremented by one. By using the table in Figure 7-5 and the waveforms in Figure 7-6, you can trace the complete operation of the 4 bit binary down counter.

Up-Down Counter. The up counting and down counting capabilities can be combined within a single counter as illustrated in Figure 7-7. AND and OR gates are used to couple the flip-flops. The normal output of each flip-flop is applied to gate 1. The complement output of each flip-flop is connected to gate 2. These gates determine whether the normal or complement signals toggle the next flip-flop in sequence. The count control line determines whether the counter counts up or down.

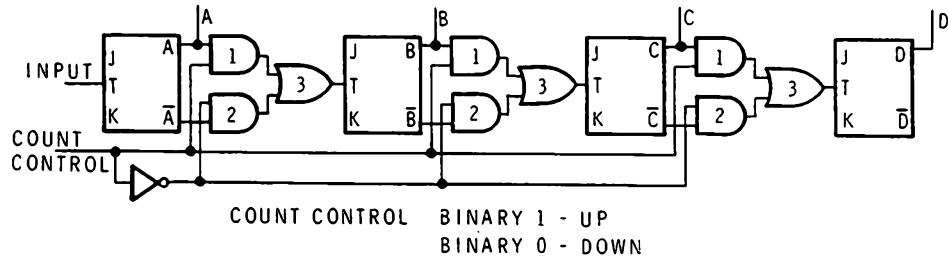


Figure 7-7
Binary up/down counter.

If the count control input is binary 1, all gate 1s are enabled. The normal output of each flip-flop then is coupled through gates 1 and 3 to the T input of the next flip-flop. The counter therefore counts up. During this time, all gate 2s are inhibited.

By making the count control line binary 0, all gate 2s are enabled. The complement output of each flip-flop is coupled through gates 2 and 3 to the next flip-flop in sequence. With this arrangement, the counter counts down.

Synchronous Counters. The counters that we have discussed so far are known as ripple counters or asynchronous counters. The term ripple is derived from the fact that the flip-flops in the counter are cascaded with the output of one driving the input of the next. As the count pulses are applied to the first or input flip-flop, the count, in effect, ripples through the flip-flops. The term asynchronous comes as a result of the flip-flops not being controlled by a single common clock pulse. Synchronous digital circuits are ones in which all elements are synchronized to a master timing signal known as a clock.

The primary advantage of a ripple counter is its simplicity. Its primary limitation is its counting speed. The counting speed of a binary counter is limited by the propagation delay of the flip-flops in the counter. In a ripple counter, the propagation delay of the flip-flops is additive. Since each flip-flop in the counter is triggered by the preceding circuit, it can

take a significant amount of time for an impulse to ripple through all of the flip-flops and change the state of the last flip-flop in the chain. This worst case condition occurs when all flip-flops in the counter change state simultaneously. Referring back to the waveforms for the four bit binary counter in Figure 7-3 you can see that this worst case condition occurs in two places. It occurs when the count changes from 0111 to 1000 and when the count changes from 1111 to 0000. If each flip-flop in the four bit circuit has a propagation delay of 50 nanoseconds, it can take as long as $4 \times 50 = 200$ nanoseconds for the D flip-flop to change state upon the application of an input pulse. Should the input pulses occur at a rate faster than 200 nanoseconds, the binary number stored in the counter will not truly represent the number of input pulses that have occurred. The counter state will lag the input signal. The upper frequency limit (f) of the ripple counter is approximately equal to

$$f = \frac{1}{nt} \times 10^9$$

where n is the number of flip-flops in the counter and t is the propagation delay time of a single flip-flop in nanoseconds.

For a flip-flop with a 35 nanosecond propagation delay in a four bit counter, the maximum counting speed is

$$f = \frac{1}{4(35)} \times 10^9 = 7.1428 \text{ MHz}$$

This means that the counter can count at speeds up to about 7 MHz without counting errors. At higher frequencies, the states of the flip-flops cannot keep pace with the rapid occurrence of the input pulses. The counter may actually lag several pulses depending upon the input frequency and the exact propagation delay. Counting errors can occur if the counting is periodically stopped and continued.

The direct solution to this problem, of course, is to use flip-flops with a lower propagation delay. Flip-flops are available to count at high frequencies up to 1GHz. The propagation delay is very small. Such flip-flops generally employ a non-saturating logic circuit such as ECL to achieve this fast counting rate. Such circuits are expensive and have high power consumption and are undesirable for many applications.

It is possible to reduce the propagation delay effects and increase the counting speed of a binary counter by using a special circuit arrangement. Counters employing this technique are known as synchronous counters.

A synchronous counter is one where all of the flip-flops are triggered simultaneously by a clock pulse or the signal to be counted. Since all flip-flops change state at the same time, the total propagation delay for the circuit is essentially equal to that of a single flip-flop. The propagation delays are not additive and therefore much higher counting speeds can be achieved.

A typical synchronous binary counter is shown in Figure 7-8. Notice that all of the flip-flop T inputs are connected together to a common count input line. This connection is what makes the counter synchronous. All of the flip-flops are synchronized to the input signal to be counted.

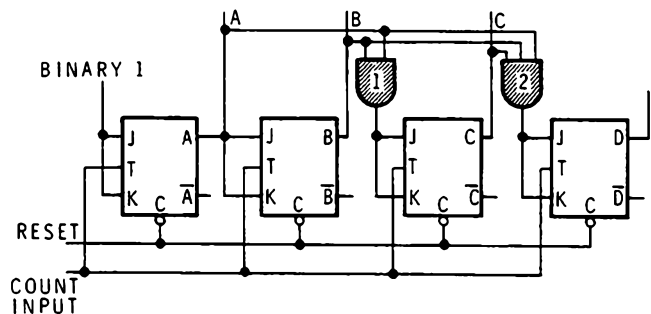


Figure 7-8
A synchronous binary counter.

The operation of the flip-flops is controlled by the states of the JK inputs. As you recall, the J and K inputs can be used as controls for the flip-flop. Up to this point in our discussion of counters we have assumed that the JK inputs were open or connected to a binary 1 level. This essentially enables the flip-flop and permits it to be toggled or complemented each time the trailing edge of an input signal appears on the T input line. If the JK inputs are brought to binary 0, the signal applied to the T input will be ignored. The flip-flop will simply remain in the state to which it was set prior to making the JK input lines low. By using the JK input lines we can then enable or inhibit the toggling of the flip-flops.

In Figure 7-8 the JK inputs on the A flip-flop are connected to binary 1 to enable the flip-flop permanently. Each time a count input signal appears the flip-flop will toggle or change state just as the A flip-flop on the ripple counters discussed earlier operated.

The JK inputs to the B flip-flop are controlled by the normal output of the A flip-flop. This means that the only time that the B flip-flop can change state is when the output of the A flip-flop is binary 1. The JK inputs to the C flip-flop are controlled by the normal outputs of both the A and the B flip-flops. The A and B signals are ANDed together in gate 1 and its output used to control the JK inputs. In order for the C flip-flop to change state, both A and B must be set. The C flip-flop will change state at the first count pulse occurring after A and B are high.

The D flip-flop is controlled by the A, B, and C flip-flops. The A, B, and C signals are ANDed in gate 2 and the output of gate 2 used to control the JK inputs.

With the circuit arrangement shown in Figure 7-8, the counting sequence is identical to that given in the Table of Figure 7-2. The waveforms of Figure 7-3 are also applicable. In other words, this circuit is still a binary counter but its mode of operation is somewhat different.

The benefit of this binary counter can be best seen by analyzing the state changes in the flip-flop. Assume that the counter contains the number 0111. This means that the A, B, and C flip-flops are set. The JK inputs to the B, C, and D flip-flops are enabled. This means that upon the occurrence of the next count input pulse, all flip-flops will toggle. When this pulse occurs, flip-flops A, B, and C will reset. The D flip-flop will be set. The new number in the counter will be 1000. The important point to note here is that all flip-flops change state simultaneously. The maximum delay between the occurrence of the count input pulse and the change of state of the outputs is only as long as the longest propagation time of the flip-flops in the circuit. All flip-flops of the same type will have approximately the same propagation delay.

Considering this same state change in the binary ripple counter, we can illustrate the effect of the accumulative propagation delay. With the number 0111 stored in the ripple counter of Figure 7-1, the output states will change as follows when an input count pulse is applied. The A flip-flop will change state first. As it does it will toggle the B flip-flop. The B flip-flop must then change state and it in turn will then toggle the C flip-flop. The C flip-flop will change state a short time later and it in turn will set the D flip-flop. Because of the finite propagation delay of each flip-flop the effect of an input count pulse does indeed ripple through the counter and it takes a specific amount of time for the correct binary number to appear in the counter.

In summary then we can say that the advantages of the synchronous counter over the ripple or asynchronous counter are as follows:

1. The synchronous counter is much faster. For the same type of flip-flop, the counting speed of the synchronous counter is significantly higher than that of the ripple counter.
2. All flip-flops in the synchronous counter change states at the same time. As the counter changes from one state to the next, there are no ambiguous states that occur because of the accumulative propagation delays.

Like the ripple counter, the synchronous counter can also be expanded to as many bits as required by the application. Each flip-flop in the counter must be controlled by all previous flip-flops in the counter through an AND gate as indicated. The higher order flip-flops will require AND gates with as many inputs as there are previous flip-flops. Keep in mind that the propagation delays of these gates while small will also have a minor effect on the counting speed of the circuit. Generally, the propagation delay of a gate is significantly lower than that of a flip-flop. This technique can also be extended to down counters as well.

Counter Control Functions. There are several common control functions that are often associated with the use of binary counters. These are reset and preset. Resetting a counter is a process of putting all of the flip-flops in the binary 0 state. In many counter applications it is necessary to clear, reset or zero the counter prior to the start of a counting operation. This process ensures that the counter starts its count sequence with no prior counts stored in the flip-flops. It ensures an accurate count of the input.

Resetting a counter is easily accomplished when JK flip-flops are used. The asynchronous clear input on the flip-flops as you recall are normally used to put the flip-flop into its binary 0 state. By bringing the clear input low, the flip-flop is reset. By connecting all of the asynchronous clear inputs together, all of the flip-flops will be reset to binary 0 simultaneously when a reset pulse is applied. Figure 7-9 shows a binary up-counter with all of the asynchronous clear inputs connected together. In order for the counter to perform normally, the reset line will rest in a high or binary 1 state. To reset the counter we momentarily bring this line to a binary 0. For typical TTL JK flip-flops, a pulse whose duration is 100 nanoseconds or more can be used to reset the counter.

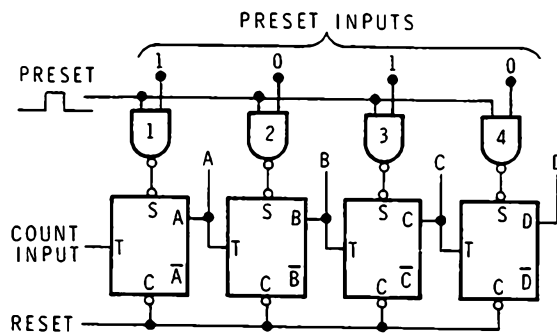


Figure 7-9
A binary counter with
reset and preset inputs.

Presetting a counter is a process of loading some binary number into the counter prior to the count sequence. It is sometimes desirable to program a counter to start counting at a particular point. The point at which the counter is to start is determined and then loaded into the counter prior to the start of the count operations.

A counter can be preset by using the asynchronous set input when JK flip-flops are used. When this input on a JK flip-flop is made binary 0, the flip-flop is set. By first clearing the counter and then setting the desired flip-flops, any binary number can be preset into the counter. The circuit shown in Figure 7-9 can be used for this purpose. In order to preset the counter to a given number, the counter is first reset by applying a binary 0 to the reset input line. Next, the desired binary number is applied to the preset inputs. There is one input for each of the flip-flops in the counter. A parallel binary number from any source can be used. When the preset input line is made high, the outputs of the gates to which the parallel input number are applied will cause the asynchronous set lines on the JK flip-flops to assume the correct states to preset the number into the counter. For example, assume that we wish to preset the number 5 into the counter. To do this we would apply the binary number 0101 to the counter as indicated. The counter is then reset, and the preset line is brought high momentarily. The parallel inputs that are binary 0 are applied to gates 2 and 4. These binary 0 inputs hold the outputs of gates 2 and 4 high regardless of the state of the preset input. This keeps the set inputs to the B and D flip-flops high. With these inputs high the flip-flops are not affected. The binary 1 states of the desired input number are applied to gates 1 and 3. When the preset input goes high, the outputs of gates 1 and 3 will go low. This will cause flip-flops A and C to become set. The number 0101 is then stored in the counter.

The method of presetting a counter shown in Figure 7-9 is somewhat awkward in that it requires two operations. First the counter must be reset and then the desired preset number is loaded. It is desirable to have the preset operation take place with a single operation. This can be accomplished by the circuits shown in Figure 7-10. Only one JK flip-flop is shown to simplify the discussion. One of these circuit arrangements would be used on each flip-flop in a counter if presetting were desired. In Figure 7-10A, gates 1 and 2 are connected to the asynchronous set and clear inputs of the JK flip-flop. The desired parallel input (IN) is applied to gate 1. A preset line is tied to both gates 1 and 2. To the input line is applied a binary 0 or binary 1 state which will specify the state of the flip-flop after the preset input is enabled. When the preset input goes high, the JK flip-flop is preset to the desired state. For example, assume that the input is binary 1. When the preset line goes high, the output of gate 1 will go low. This will cause the set(S) input of the JK flip-flop to go low thereby setting the flip-flop. The low on the output of gate 1 will cause the output of gate 2 to remain high. This has no effect on the C input. A low input to gate 1 will reset the flip-flop. With a low input the output of gate 1 will be high. This high output does not affect the S input to the JK flip-flop. It does however enable gate 2. When the preset line goes high, the output of gate 2 will go low. This causes the JK flip-flop to be put in a binary 0 state. The preset operation takes place with only the single operation of applying the preset input pulse. Note that since the asynchronous inputs are used, the presetting operation will over-ride all other flip-flop operations.

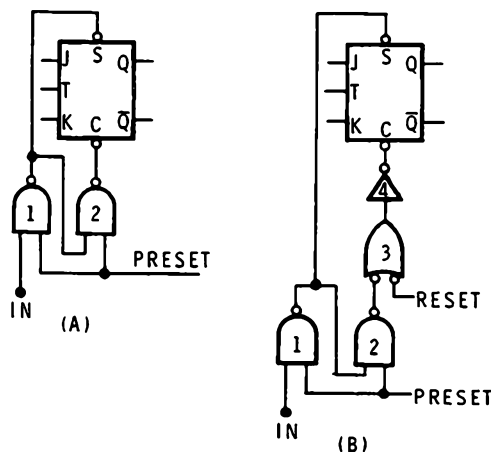


Figure 7-10
Methods of presetting a counter
(A) preset only (B) preset with reset.

Sometimes it is desirable to combine both the reset and preset functions in a counter. This can be accomplished with the circuit shown in Figure 7-10B. Here gates 1 and 2 perform the same basic operation as they do in the circuit of Figure 7-10A. They are used to preset the flip-flops to the desired state. Gate 3 and inverter 4 provide an ORing function to permit the reset function to be incorporated. If the reset input line is brought low, the output of gate 3 will go high and the output of inverter 4 will go low. This will reset the flip-flop. This occurs regardless of the conditions of the preset inputs.

To preset the flip-flop, the desired binary state is applied to the input line on gate 1. When the preset line is brought high, the flip-flop will be put into the states specified by the input. The operation is identical to the circuit in Figure 7-10A. The important point to note about this circuit is that the reset and preset operations are independent of one another. It is not necessary to reset the flip-flop prior to presetting it as was the case in the circuit of Figure 7-9.

An important point to remember is that these circuits for resetting and presetting flip-flops can be applied to any type of counter: synchronous, asynchronous, up-counter, down-counter, binary, or BCD counter.

Typical Integrated Circuit Counters

While it is still sometimes necessary and desirable to implement counters with individual IC JK flip-flops, most counter applications can be met with a variety of available MSI integrated circuit counters. All of the most often used binary counters have been implemented in integrated circuit form thereby eliminating the necessity for designing such counters for each application. A variety of counter types and specifications are available. In designing digital equipment, it is desirable to first investigate the types of MSI IC counters available. In most cases you will find one suitable for your application. Only in rare cases where an unusual or peculiar type of counter for unique applications is required will it be necessary for you to design a special counter. However, for such applications, versatile integrated circuit JK flip-flops are available.

In this section we are going to consider one of the most popular and widely used integrated circuit binary counters available.

Figure 7-11 shows the logic diagram of the 74193 TTL MSI IC counter. This is a four bit synchronous counter that can be used for either up or down counting. It also has separate clear or reset inputs as well as the ability to be preset from some external four bit parallel source. In other words, this particular device incorporates all of the features we have considered in a binary counter up to this point.

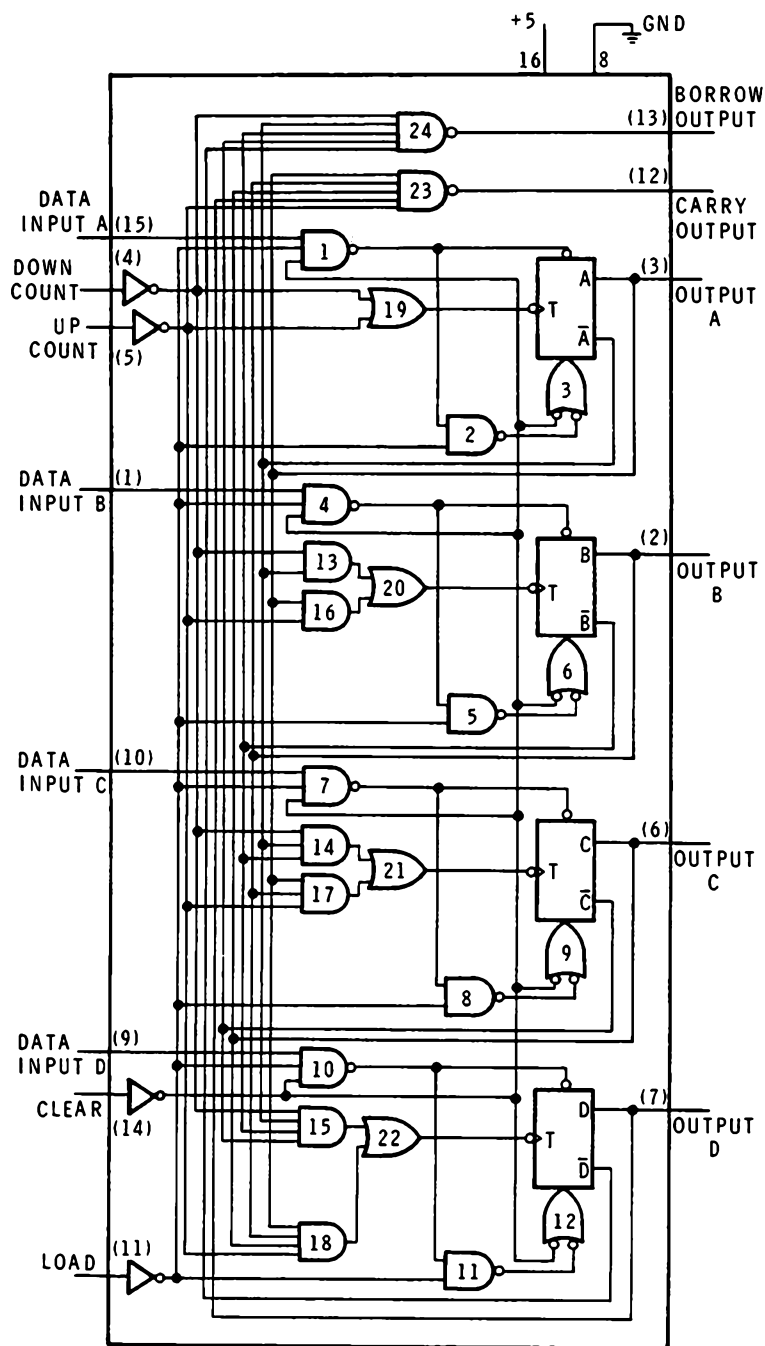


Figure 7-11
Type 74193 TTL MSI binary counter.

As shown in the diagram in Figure 7-11, the counter consists of four JK flip-flops. Gates 1 through 12 make up the logic circuitry used in the reset and preset operations. This circuitry is similar to the reset and preset circuit operations discussed previously. To reset this counter, you apply a high or binary 1 level to the clear input line. This forces all four flip-flops into the binary 0 state. The clear operation is asynchronous and overrides all other counter functions.

The counter can be preset by applying a parallel 4-bit binary number to the data inputs. Data input A is the LSB. When the load input is brought low, the 4-bit input number will be loaded into the flip-flops. This preset function is also asynchronous and will override any synchronous counting functions that occur.

The input pulses to be counted are applied to either the up-count input or the down-count inputs. Instead of having a single count input and an up/down control line as described in the previous discussion of up/down counters, this IC counter uses two separate inputs. To increment the counter, pulses are applied to the up-count input. To decrement the counter, pulses are applied to the down-count input. The counter changes state on the leading edge of the applied input pulse. In other words, it is the binary 0-to-binary 1 transition at the count input that causes the counter to change state. In order for the counter to operate properly, the unused count input must be in the high or binary 1 state while count pulses are applied to the other input. The up and down count sequences are identical to those considered previously.

Synchronous operation is used in this counter by having the input count pulses clock all flip-flops simultaneously so that all outputs change coincidentally with one another. Instead of controlling each flip-flop by use of the JK inputs as in the previously discussed synchronous counter, gates are used ahead of the T inputs to the flip-flops for this purpose. The outputs of the flip-flops control the states of the gates ahead of the T inputs to permit the count pulse to be applied at the appropriate time. Gates 16, 17, and 18 are used to control the application of the up count pulses to the T inputs. The up count input is applied to these gates simultaneously. Note that the outputs of the previous flip-flops are connected to the inputs of these gates in order to control when the count pulse is allowed to toggle the flip-flop. Gate 13, 14, and 15 perform the same function for the down-count operation. Gates 19, 20, 21, and 22 are simply OR gates that permit either the up or down count pulses to appear at the flip-flop T inputs.

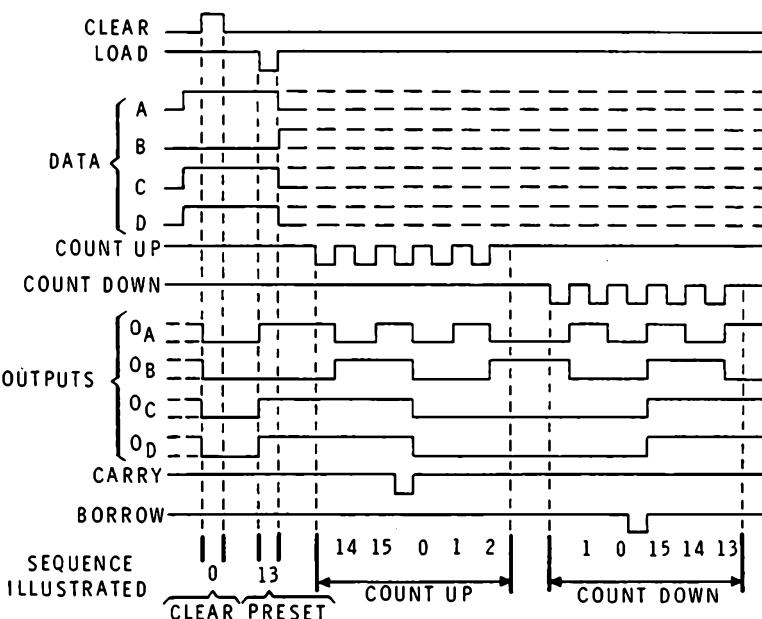
This counter has both carry and borrow output gates which are used for cascading these counters. When it is necessary to use a counter with more than 16 states, several of these ICs can be cascaded to provide counters whose lengths are some multiple of 4.

The carry output is developed by gate 23. This NAND gate monitors the normal outputs of the flip-flops. When all of the normal outputs are binary 1 and the up-count pulse occurs, the carry output line will go low. The duration of the carry output pulse is equal to the duration of the count input pulse. This pulse indicates that the counter is in its maximum count state (1111) and that the next count input pulse will cause it to recycle to 0000. This carry output pulse is connected to the count up input of the next counter in sequence when counters are cascaded.

The borrow output is produced by gate 24. This gate monitors the complement outputs of the four flip-flops. The down count input is also applied to gate 24. When the counter has been decremented to the 0000 state and with the down count input high, the output of gate 24 will go low. This indicates the counter is in its lowest count state (0000) and that upon application of the next count input pulse it will recycle to 1111. To cascade 74193 counters for down counting applications, the borrow output is connected to the down count input of the next counter in series.

As you can see, this device is a very flexible counting unit. It can perform nearly any of the required basic counting functions often encountered in digital work. Figure 7-12 illustrates the operation of this counter. The waveforms for the clear, preset, count up, and count down operations are shown.

Figure 7-12
Typical clear, load, and
count sequences for the 74193 counter.



Self Test Review

1. In a binary counter using JK flip-flops, the counter state will change when the T input changes from
 - a. high to low
 - b. low to high
 - c. both a. and b.
2. A four bit binary counter contains the number 0100. Nine input pulses occur. The new counter state is:
 - a. 0010
 - b. 1001
 - c. 1011
 - d. 1101
3. A four bit binary counter contains the number 1010. Seven input pulses are applied. The new counter state is:
 - a. 0001
 - b. 0101
 - c. 1100
 - d. 1111
4. A binary counter constructed with two 74193 ICs has a maximum count capability of
 - a. 15
 - b. 16
 - c. 255
 - d. 256
5. A binary counter made up of 5 JK flip-flops will divide an input frequency by
 - a. 5
 - b. 8
 - c. 16
 - d. 32
6. Input pulses applied to a down counter cause it to be _____.
7. Clearing a counter to zero is known as _____.
8. Setting a counter to a desired state is called _____.

9. A four bit down counter is in the 0110 state. Fourteen input pulses occur. What is the new output state?
 - a. 0110
 - b. 0100
 - c. 1000
 - d. 1110
10. The borrow output on the 74LS193 counter detects the counter state _____.
11. An asynchronous counter is faster than a synchronous counter assuming the same flip-flops are used to implement both.
 - a. True.
 - b. False.
12. The state of a binary counter is determined by monitoring the _____ outputs of the flip-flops.
13. The state of the input or first flip-flop in a counter represents the _____ of the number stored in the counter.
14. Synchronous operation of the flip-flops in the 74LS193 counter is obtained by controlling the
 - a. JK inputs
 - b. T inputs
 - c. direct set and clear inputs.
15. In the 74LS193 IC counter, the counter is incremented or decremented by the
 - a. leading edge
 - b. trailing edgeof the input pulse. The counter is reset by a
 - c. binary 0
 - d. binary 1on pin 14. The counter is preset by a
 - e. binary 0
 - f. binary 1on pin 11.

Answers

1. a. high to low (1 to 0)
2. d. 1101 ($4 + 9 = 13$)
3. a. 0001 ($10 + 7 = 17$) The maximum count capability of a 4 bit counter is 15. With 10 in the counter initially, it will reach maximum counter (1111) after the fifth input is applied. The sixth input pulse will recycle the counter to 0000. The seventh input pulse will put the counter into the 0001 state.
4. c. 255. Each 74LS193 has four flip-flops.
 $2^8 - 1 = 256 - 1 = 255$
5. d. $32 \cdot 2^n = 2^5 = 32$
6. decremented
7. resetting
8. presetting
9. c. 1000 The first six input pulses, decrement the counter to 0000. The seventh pulse recycles the counter to 1111. The next seven pulses decrement the counter to 1000.
10. 0000
11. b. False. Synchronous counters are always faster than asynchronous counters.
12. normal
13. LSB
14. b. T inputs. Gates ahead of the T input controlled by the flip-flop states determine when the flip-flops toggle.
15. a. leading edge (0 to 1 transition)
d. binary 1
e. binary 0

BCD COUNTERS

A BCD counter is a sequential circuit that counts by tens. The BCD counter has ten discrete states which represent the decimal numbers 0 through 9. Because of its ten state nature, a BCD counter is also sometimes referred to as a decade counter.

D	C	B	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

← RECYCLE

Figure 7-13
Count sequence of
8421 BCD counter.

The most commonly used BCD counter counts in the standard 8421 binary code. The table in Figure 7-13 shows the count sequence. Note that a four bit number is required to represent the ten states 0 through 9. These ten four bit codes are the first ten of the standard pure binary code. As count pulses are applied to the binary counter, the counter will be incremented as indicated in the table. Upon the application of a tenth input pulse, the counter will recycle from the 1001 (9) state to the 0000 state.

An asynchronous 8421 BCD counter constructed with JK flip-flops is shown in Figure 7-14. This counter will generate the BCD code given in the Table of Figure 7-13. Note that the counter consists of four flip-flops like the four bit pure binary counter discussed earlier. The output of one flip-flop drives the T input to the next in sequence, thereby, making this BCD counter a ripple or asynchronous type. Unlike the binary counter discussed earlier, however, this circuit has several modifications which permits it to count in the standard 8421 BCD sequence. The differences consist of a feedback path from the complement output of the D flip-flop back to the J input of the B flip-flop. Also a two input AND gate monitors the output states of flip-flops B and C and generates a control signal that is used to operate the J input to the D flip-flop. These circuit modifications in effect *trick* the standard four bit counter and cause it to recycle every ten input pulses.

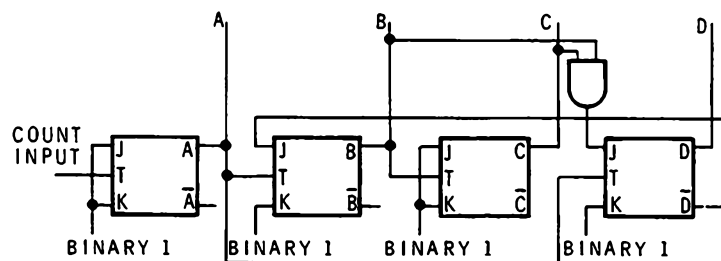


Figure 7-14
An asynchronous 8421 BCD counter.

The waveforms shown in Figure 7-15 illustrate the operation of the 8421 BCD counter. The count sequence is identical to that of the standard four bit pure binary counter discussed earlier for the first 8 input pulses. The operations that occur during the 9th and 10th pulses are unique to the BCD counter.

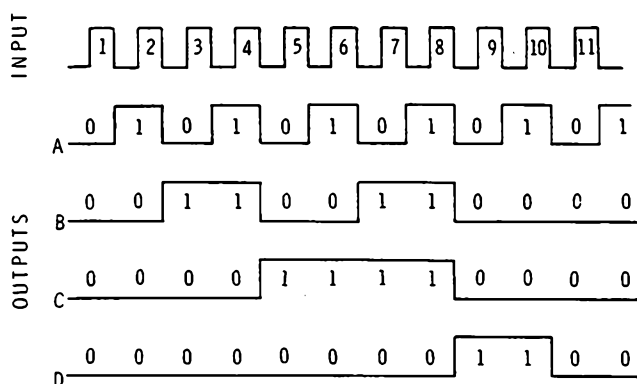


Figure 7-15
Waveforms of the 8421 BCD counter.

Assume that the counter in Figure 7-14 is initially reset. The outputs of flip-flops B and C will be binary 0 at this time. This makes the output of the AND gate low and causes the J input of the D flip-flop to be held low. The D flip-flop cannot be set by the toggle input from the A flip-flop until the J input goes high. Note also that the complement output of the D flip-flop which is binary 1 during the reset state is applied to the J input of the B flip-flop. This enables the B flip-flop permitting it to toggle when the A flip-flop changes state.

If count pulses are now applied, the states of the flip-flops will change as indicated in Figure 7-15. The count ripples through the first three flip-flops in sequence as in the standard 4 bit binary counter considered earlier. However, consider the action of the counter upon the application of the 8th input pulse. With flip-flops A, B, and C set and D reset, the B and C outputs are high thereby enabling the AND gate and the J input to the D flip-flop. This means that upon the application of the next count input that all flip-flops will change state. The A, B, and C flip-flops will be reset while the D flip-flop is set. The counter state changes from 0111 to 1000 when the trailing edge of the 8th input pulse occurs.

In this new state, the B and C outputs are low therefore causing the J input to the D flip-flop to again be binary 0. With the J input 0 and the K input binary 1 and the D flip-flop set, the conditions are right for this flip-flop to be reset when the T input switches from binary 1 to binary 0. In addition, the complement output of the D flip-flop is low at this time thereby keeping the J input to the B flip-flop low. The B flip-flop is reset at this time and therefore the occurrence of a clock pulse at the T input will not affect the B flip-flop.

When the 9th input pulse occurs, the A flip-flop sets. No other state changes occur at this time. The binary number in the counter is now 1001. The transition of the A flip-flop switching from binary 0 to binary 1 is ignored by the T input of the D flip-flop.

When the 10th input pulse occurs, the A flip-flop will toggle and reset. The B flip-flop will not be affected at this time since its J input is low. No state change occurs in the C flip-flop since the B flip-flop remains reset. The changing of the state of the A flip-flop however, does cause the D flip-flop to reset. With its J input binary 0 and K input binary 1, this flip-flop will reset when the A flip-flop changes state. This 10th input pulse therefore causes all flip-flops to become reset. As you can see by the waveforms in Figure 7-15, the counter recycles from the 1001 (9) state to the 0000 state on the 10th input pulse.

Numerous variations of the basic BCD counter in Figure 7-14 are possible. Using the same basic count modifying techniques, a synchronous BCD counter can be constructed. All of the flip-flops are toggled simultaneously by the common count input. As in the binary counter, the counting speed of the BCD counter can be significantly increased by this synchronous technique. In addition, it is also possible to construct a BCD down counter. Each time an input pulse is applied, the BCD counter is decremented. The count sequence is from 9 through 0.

Cascading BCD Counters. A single BCD counter has a maximum of ten discrete states and therefore can only represent the number 0 through 9. When the counter must count more than ten pulses, several BCD counters must be cascaded. Each BCD counter in the counting chain will represent one decimal digit. The number of BCD counters used determines the maximum count capabilities.

Figure 7-16 shows a counter chain with four BCD counters. Each BCD counter is represented by a single block to simplify the drawing. The count input line and the four flip-flop output lines are designated for each counter. In each case, the A output is the least significant bit and the D output is the most significant bit of that counter. The input BCD counter contains the least significant digit of the count contained in the counter. The most significant digit is represented by the counter on the far right. Since this counter contains four BCD counters, the maximum count capability is 9999.

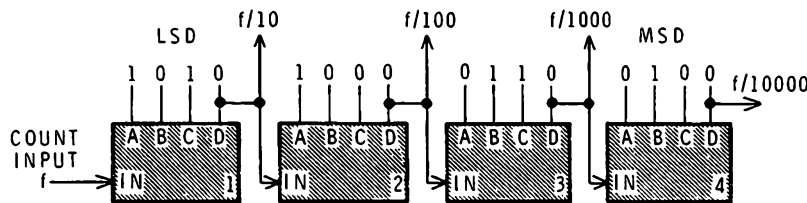


Figure 7-16
Cascading BCD counters to
increase count capability.

As count input pulses are applied to BCD counter number 1, it will be incremented as indicated previously. The output states will change in accordance with the 8421 BCD code. Note that the most significant bit output (D) of this first counter is connected to the count input of the second BCD counter. Each time the input counter counts ten pulses and recycles, it will trigger the next counter in sequence. By referring back to the waveforms in Figure 7-15, you can see that the trailing edge of the D output occurs on the trailing edge of the 10th input pulse. As this 10th input pulse occurs, the input counter recycles to 0 and the trailing edge increments the next counter in sequence to 1. The remaining counters in the chain are connected in the same way. As you can see then the counter does perform a decimal counting function with each BCD counter representing one of the decimal digits. The decimal contents of the counter can be determined by observing the flip-flop outputs. In Figure 7-16, the counter contains the decimal number 2615. This means that 2615 pulses have occurred at the input assuming the counter was initially reset.

The BCD Counter as a Frequency Divider. Like any counter, the BCD counter can also be used as a frequency divider. Since the BCD counter has ten discrete states, it will divide the input frequency by ten. The output of the most significant bit flip-flop in the BCD counter will be one tenth of the input frequency. From Figure 7-15, you can see that only a single output pulse occurs at the D output for every ten input pulses. While the D output does not have a 50 percent duty cycle, the frequency of the signal is nevertheless one tenth of the input frequency.

By cascading BCD counters, the input frequency can be reduced by any desired factor of ten. For example, in the counter of Figure 7-16, the output of the 4th BCD counter will be $\frac{1}{10000}$ th of the input frequency. The output of the third counter will be $\frac{1}{1000}$ th of the input frequency. The output of the second, of course, will be $\frac{1}{100}$ th of the input frequency. If an input signal of 2 MHz is applied to the counter in Figure 7-16, the D output of the MSD counter will be 200 Hz. When used as a frequency divider, the BCD counter is often referred to as a decade scaler.

Typical Integrated Circuit BCD Counter. The most widely used integrated circuit BCD counter is the type 74LS90A. This TTL MSI counter is an asynchronous or ripple counter that counts in the standard 8421 BCD code. The logic diagram of this counter is shown in Figure 7-17. Logically, it is identical to the BCD counter discussed earlier. The counter is made up of four JK flip-flops and the associated gating to permit the 8421 BCD sequence.

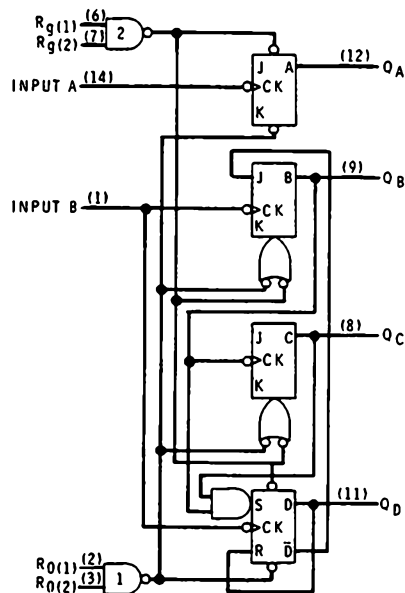


Figure 7-17

Logic diagram of 74LS90A BCD counter.

A look at the logic diagram of the 74LS90A counter in Figure 7-17 shows that the A flip-flop is not internally connected to the other three flip-flops. In order to produce an 8421 BCD count, the A output must be connected to the B input. This must be done externally. The input pulses to be counted then are applied to input A.

Gate 1 in Figure 7-17 is used to reset the flip-flop. When both inputs to gate 1 are high, all four flip-flops in the counter will be put in the binary 0 state. This permits two or more inputs to control the resetting of the flip-flop. Normally, only one input will be necessary and therefore both of the reset inputs can be simply tied together.

Gate 2 in the 74LS90A counter is used to preset the counter to the binary state 1001 or 9. When both inputs to gate 2 are high, a 9 will be preset into the counter. This particular function is useful in applications requiring arithmetic operations to be performed with BCD counters.

Despite the fact that the 74LS90A counter is an asynchronous counter, its maximum count frequency is approximately 32 MHz. This TTL MSI counter comes in a 14 pin DIP and is widely used in scaling and counting applications.

Self Test Review

16. A BCD counter can assume _____ discrete states.
17. A BCD counter is in the 0111 state. How many input pulses were applied to it after it was reset?
 - a. 3
 - b. 6
 - c. 7
 - d. 12
18. A BCD counter divides its input signal frequency by _____.
19. A 74LS90A IC is preset to 1001. Six count pulses are then applied. What is the counter state?
 - a. 0000
 - b. 0101
 - c. 0110
 - d. 1001
20. The BCD counter in Figure 7-16 has the following outputs.
(1.) 1001 (2.) 0010 (3.) 1000 (4.) 0101
How many input pulses does this represent? _____.
21. If a 5 MHz signal was applied to the BCD counter of Figure 7-16, the output of counter 3 would be:
 - a. 500 Hz
 - b. 5 kHz
 - c. 50 kHz
 - d. 500 kHz
22. When used as a frequency divider, the BCD counter is referred to as a _____.
23. A chain of 6 decade counters has a maximum count capacity of _____.
24. A BCD counter is cascaded with a 3 flip-flop binary ripple counter. The overall frequency division ratio is:
 - a. 20
 - b. 30
 - c. 60
 - d. 80

Answers

- 16. 10
- 17. c. 7
- 18. 10
- 19. b. 0101 The first input pulse recycles the counter from 1001 to 0000. The next five pulses increment it to 0101.
- 20. 5829 (counter 4 is the MSD.)
- 21. b. 5 kHz Each counter divides by 10. The third counter in the chain has an output that is 10^3 lower than the input of 5 MHz or $\div 1000 = \text{kHz}$.
- 22. decade scaler.
- 23. 999999
- 24. d. 80 The BCD counter divides by ten. The 3 flip-flop binary ripple counter divides by $2^3 = 8$. The total division is the product of the two dividers or $8 \times 10 = 80$.

SPECIAL COUNTERS

Binary and BCD counters are by far the most commonly used counters in digital systems. Most counting applications can be implemented with MSI binary and BCD counters. However, there are some applications where a special counter may be required. It may be necessary to count in a peculiar sequence or to have the counter sequence through the states of some special code. In other applications it may be desirable to divide or scale an input frequency by some value other than even powers of two or ten. Special counters can be constructed to perform all of these applications. Some typical examples are a counter that counts in the Gray code or a frequency divider that divides the input by seven.

Because of the flexibility of the JK flip-flop, such special counters are relatively easy to implement. The basic approach is to construct a standard binary counter and then by using feedback and input gating controls on the JK inputs a counter can be developed to count in any sequence with as many individual states as desired.

We refer to the number of discrete states that a counter can assume as the modulus of that counter. A modulo N counter is one that has N states. A BCD or decade counter is a modulo 10 counter since it can assume one of ten discrete binary states. The binary counters that we discussed earlier are modulo N counters where N is some power of two. A counter containing four flip-flops is a modulo 16 counter. As you have seen it is easy to construct a binary counter whose modulo is some power of two. BCD counters with a modulo of 10 are also easily assembled. However, there are other applications that require counters with modulos of other integer values.

Modulo 3 Counter. A modulo three counter is shown in Figure 7-18. Since the T inputs to both JK flip-flops are connected to the count input, the circuit is synchronous. The feedback line from the \bar{B} output back to the J input of the A flip-flop causes the circuit to count by three.

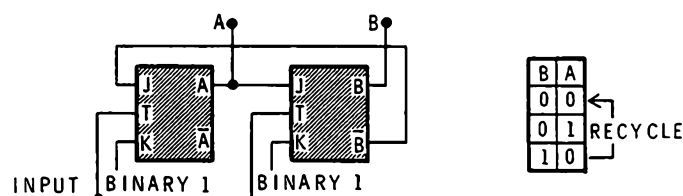


Figure 7-18
A modulo 3 counter
and its count sequence.

To determine the operation of the counter assume that both flip-flops are initially in the reset state. The A and B outputs are binary 0. The low output of flip-flop A holds the J input to flip-flop B low. When the trailing edge of the first input pulse occurs, flip-flop A will set. The \bar{B} output is holding the J input to A flip-flop high thereby enabling it to be set when the proper T input pulse occurs. When the first input pulse occurs, A is set. The B flip-flop is not affected. With the A output high, the J input to the B flip-flop is now high thereby permitting that flip-flop to toggle upon application of the next input pulse. The \bar{B} output of the B flip-flop is still high thereby continuing to enable the J input of the A flip-flop.

When the trailing edge of the second input pulse occurs, the A flip-flop will again toggle. This time it will reset. At the same time the B flip-flop will set. The J input to the B flip-flop is now low while the J input to the A flip-flop is also low. When the next clock pulse occurs, the B flip-flop will reset. The A flip-flop having been previously reset simply remains reset. On the application of the trailing edge of the third input pulse, the counter state cycles back to its original reset condition. If further input pulses are applied, the counter will simply repeat the cycle just described. The count sequence for the modulo three counter is shown in Figure 7-18. The waveforms showing the operation of the modulo three counter are illustrated in Figure 7-19. Trace through the operation of the circuit using these waveforms as a guide to be sure you fully understand how the circuit operates. This circuit can be used for simple counting applications requiring a three state counter. The circuit can also be used as a divide by 3 scaler. The output of either the A or B flip-flops has a frequency that is one third of the count input frequency. A single output pulse occurs for every three input pulses as indicated by the waveforms in Figure 7-19.

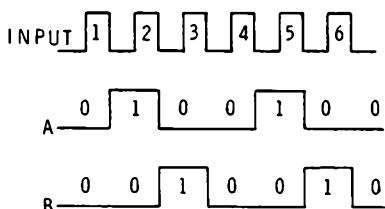


Figure 7-19
Waveforms for the modulo 3 counter.

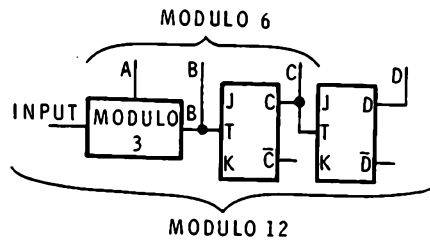


Figure 7-20

Forming modulo 6 and modulo 12 counters using a modulo 3 circuit as a base.

By cascading a modulo three counter with additional JK flip-flops, modulo six and modulo twelve counters are easily formed. This is shown in Figure 7-20. By connecting a JK flip-flop to the output of the modulo three counter, a modulo six counter is formed. The modulo three will cycle through its three states twice, once with the JK flip-flop C reset and again with the JK flip-flop C set. This produces a total of six discrete states as indicated in the table shown in Figure 7-21A. If you will look closely at the sequence of the states you will find that this does not correspond to the standard binary code. Since the code produced is not the pure binary code or the BCD code, it is referred to as an unweighted code.

C	B	A
0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
1	1	0

(A)

RECYCLE

COUNT SEQUENCE
MODULO 6 COUNTER

D	C	B	A
0	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	0	1
1	1	1	0

(B)

RECYCLE

COUNT SEQUENCE FOR
MODULO 12 COUNTER

Figure 7-21

Count sequence for a (A) modulo 6 counter and a (B) modulo 12 counter.

By adding another JK flip-flop (flip-flop D) as shown in Figure 7-20, a modulo 12 counter is formed. Here the modulo 6 counter cycles through its six states two times, once while the D flip-flop is reset and again while it is set. This produces the 12 discrete states shown in Figure 7-21B. The counter shown in Figure 7-20 can be used to perform frequency division by three, six, or twelve.

Counters with a modulo 6 and a modulo 12 can also be formed by putting the JK flip-flops ahead of the modulo three counter instead of after it. The counter thus formed still has six or twelve states respectively. However, the binary code produced by this arrangement is different from that obtained by the connection shown in Figure 7-20. When using this arrangement in frequency divider applications, it generally doesn't matter which connection is used as the output frequency will always be either $\frac{1}{6}$ th or $\frac{1}{12}$ th of the input frequency. Where the specific code sequence is critical however, these two arrangements should be carefully considered. It should be pointed out that the counter shown in Figure 7-20 produces a 50 percent duty cycle when the C and D outputs are used. By putting the modulo 3 counter after the cascaded JK flip-flops, a duty cycle other than 50 percent will be produced.

Modulo 5 Counter. A modulo five counter is shown in Figure 7-22. The counter produces five distinct three bit states. Synchronous operation is obtained by applying the count input to the T input of all three JK flip-flops. A combination of feedback and external logic gates are used to cause the counter to count by five.

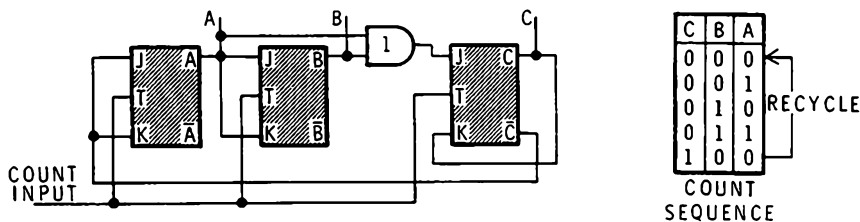


Figure 7-22
Modulo 5 counter and its count sequence.

The count sequence for this circuit is shown in Figure 7-22. Note that the count sequence simply recycles for the application of each five input pulses.

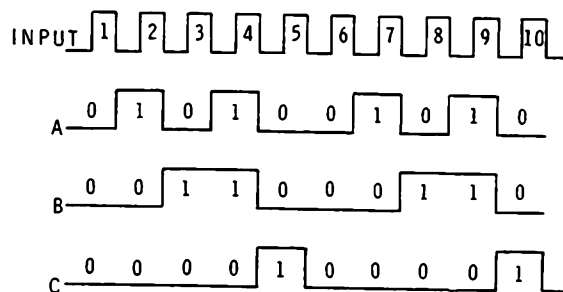


Figure 7-23
Waveforms for the modulo 5 counter.

The input and output waveforms of the modulo five counter are illustrated in Figure 7-23. Compare the output states of these waveforms for each of the flip-flops to the count sequence table in Figure 7-22.

The detailed operation of this counter is not included here. It will be excellent practice for you to reason out the count sequence of this circuit yourself. Remembering the operation of the JK flip-flop and using the count sequence table and the waveforms in Figure 7-23, trace the operation of the counter for all five states until it recycles. As a starting point, assume that all three flip-flops are initially reset.

If you will refer back to Figure 7-17 showing the diagram of the type 74LS90A decade counter you will see that flip-flops B, C, and D in this circuit form a modulo 5 counter by themselves. As indicated earlier a separate input is used for this circuit. Normally, it is tied to the output of the A flip-flop to produce BCD counting. However, the modulo 5 section of this counter can be used independently by simply applying the count input to the B terminal. The A flip-flop is not used.

Modulo N Counters with MSI. While JK flip-flops can be interconnected by the use of feedback and external logic gates to form a counter with any desired modulo, the availability of MSI integrated circuit counters greatly simplifies the construction of modulo N counters. The type 74LS193 TTL MSI synchronous up/down counter discussed earlier is an excellent choice for implementing modulo N counters.

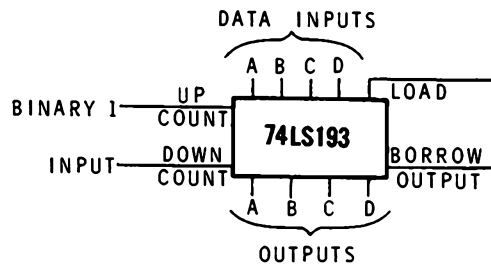


Figure 7-24
A type 74LS193 TTL MSI
Counter used as a modulo N counter.

Figure 7-24 shows the type 74LS193 TTL MSI counter connected as a modulo N counter. The counter is connected to count down. For frequency divider applications, it does not matter whether the counter counts up or down. The up count input is held to a binary 1 while the input pulses are applied to the down count input. The borrow output line which essentially detects the 0000 state is connected back to the load input line of the counter. To the four parallel data input lines is connected a binary word that will determine the modulus of the counter. The modulus of the counter is equal to the binary equivalent of the decimal number applied to the data inputs. This is indicated by the table in Figure 7-25. For example to obtain a modulo 7 counter, the binary number for the decimal number seven (0111) is connected to the data inputs.

In operation, the counter is preset to the binary number applied to the data inputs. The counter is then decremented by the input pulses. It down counts in binary until the zero state is reached. At this time the borrow output line goes low and again presets the counter to the number applied to the data inputs. This sequence repeats as long as pulses are applied to the count input.

MODULO	DATA INPUTS			
	D	C	B	A
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Figure 7-25
Parallel data input code
and related decimal modulus.

As soon as the borrow output line goes low, the binary number applied to the data inputs will be immediately (asynchronously) loaded into the counter. Of course as soon as the new number is loaded into the counter, the borrow output will disappear since the counter state will no longer be 0000. What this means is that the duration of the borrow output pulse must be long enough in order to ensure that the data input is loaded before the zero output state disappears. In order to ensure that this condition happens, the input duration of the clock pulse must be greater than the total propagation delay of the gates in the counter associated with presetting the number on the data inputs. Recall from the previous discussion of the operation of the 74LS193 counter that the borrow output is also derived from the down count input signal.

Even though the borrow output pulse disappears as soon as the counter is preset to the data input states, the internal propagation delays of the circuit are such that all flip-flops become preset before the borrow output disappears. The load input signal must propagate through both the flip-flops and the gates in the circuit. Since the propagation delay through the various circuits in the counters vary from one device to the next, it is possible that erratic operation can occur if the propagation delays are too short. The reliability of the counter can therefore be improved by adding some propagation delay between the borrow output and the load input pins. This can be accomplished by cascading a number of inverters between these two pins on the circuit. Be sure to use an even number of inverters so the proper polarity binary signal will be applied.

Self Test Review

25. Determine the output frequency of the circuit shown in Figure 7-26.
26. What is the overall circuit modulo? _____.

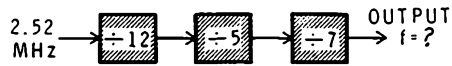


Figure 7-26
Circuit for self test review question 25.

26. What is the modulo of the circuit in Figure 7-27? _____
Sketch the input and output waveforms and make a table of the count sequence.

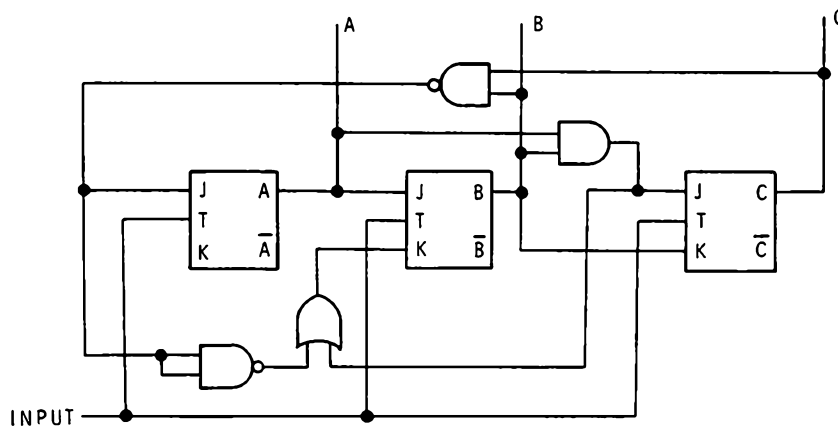


Figure 7-27
Circuit for self test review question 26.

Answers

25. 6 kHz modulo 420 ($12 \times 5 \times 7$)

26. modulo 7 See Figure 7-28

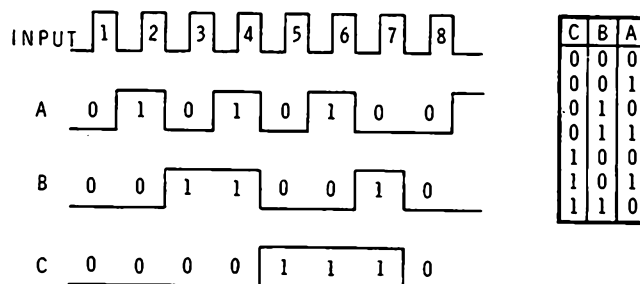


Figure 7-28
Waveforms and count sequence
for modulo 7 counter in Figure 7-27.

NOTE: Using 2-74LS76 JK flip-flop ICs and a 74LS00 IC, you can breadboard the modulo 7 counter in Figure 7-28 on the Experimenter and verify its operation.

SHIFT REGISTERS

Another widely used type of sequential logic circuit is the shift register. Like a counter, a shift register is made up of binary storage elements. While flip-flops are the most commonly used storage element in shift registers, other types of circuits are also used. The storage elements in a shift register are cascaded in such a way that the bits stored there can be moved or shifted from one element to another adjacent element. All of the storage registers are actuated simultaneously by a single input clock or shift pulse. When a shift pulse is applied, the data stored in the shift register is moved one position in one of two directions. The shift register is basically a storage medium where one or more binary words may be stored. However, because of the ability to move the data one bit at a time from one storage element to another makes the shift register valuable in performing a wide variety of logic operations.

Shift Register Operation

The illustration in Figure 7-29 shows how a shift register operates. Here the shift register consists of four binary storage elements such as flip-flops. The binary number 1011 is currently stored in the shift register. Another binary word, 0110, is generated externally and is available to the shift register serially. As shift pulses are applied, the number stored in the register will be shifted out and lost while the external number will be shifted into the register and retained.

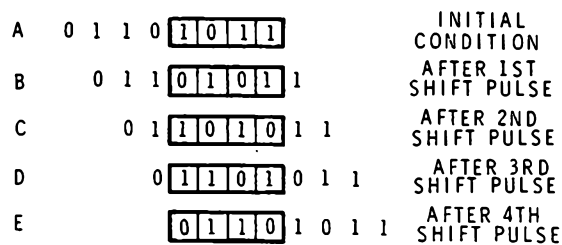


Figure 7-29
Operation of a shift register.

The initial conditions for this shift register are illustrated in Figure 7-29A. After one clock pulse, the number stored in the register initially is shifted one bit position to the right. The right most bit is shifted out and lost. At the same time, the first bit of the externally generated serial number is shifted into the left most position of the shift register. This is illustrated in Figure 7-29B. The remaining three illustrations in C, D, and E show the results after the application of additional shift pulses. After four shift pulses have occurred, the number originally stored in the register has been completely shifted out and lost. The serial number appearing at the input on the left has been shifted into the register and now resides there.

This figure illustrates several important points about a shift register. First, it indicates that the basic shift register operations are serial in nature. That is, data is moved serially, a bit at a time, into and out of the register. Most shift register operations are serial operations but many circuits are provided with both parallel inputs and parallel outputs. Such shift registers permit data to be preset in parallel and data to be read out in parallel. The ability to combine both serial and parallel operations makes the shift register an ideal circuit for performing serial to parallel and parallel to serial data conversions.

Another important point to note is that the data is shifted one bit position for each input clock or shift pulse. Clock pulses have full control over the shift register operation. In this shift register, the data was shifted to the right. However, in other shift registers it is also possible to shift data to the left. The direction of the shift is determined by the application. Most shift registers are of the shift right type.

The shift register is one of the most versatile of all sequential logic circuits. It is basically a storage element used for storing binary data. A single shift register made up of many storage elements can be used as a memory for storing many words of binary data. Such memories are referred to as serial memories since the data stored in them is entered and removed in serial form.

Shift registers can also be used to perform arithmetic operations. Shifting the data stored in a shift register to the right or to the left a number of bit positions is equivalent to multiplying or dividing that number by a specific factor. As indicated earlier, the shift register is also widely used for serial to parallel and parallel to serial data conversions. Shift registers can also be used for generating a sequence of control pulses for a logic circuit. And in some applications shift registers can be used to perform counting and frequency dividing.

In this section you are going to study the basic operation of a shift register. One of the most commonly used types of shift registers is the bipolar shift register, which is made up of flip-flops. These can be constructed with individual JK flip-flops or are available in a variety of configurations in MSI form. Another type of shift register is the MOS shift register. These registers made with MOSFETs are available in two basic types, static and dynamic. Static shift registers are made up of MOSFET flip-flops. Dynamic shift registers are made up of storage elements that take advantage of the unique characteristics of MOSFETs, namely their high impedance and capacitive nature. Because of the small size of the MOSFET structure, many storage elements can be made on a single chip of silicon. Therefore, long shift registers capable of storing many words can be made very small and economical. Both types of shift registers are widely used in digital systems.

Bipolar Logic Shift Registers

Shift registers constructed from bipolar logic circuits such as TTL and ECL circuits are usually implemented with JK flip-flops. Type D flip-flops can also be used, but shift registers implemented with JK flip-flops are far more versatile. A typical shift register constructed with JK flip-flops is shown in Figure 7-30. The serial input data and its complement are applied to the JK inputs of the input (A) flip-flop. From there the other flip-flops are cascaded with the outputs of one connected to the JK inputs of the next. Note that the clock (T) input lines to all flip-flops are connected together. The clock or shift pulses are applied to this line. Of course, since all flip-flops are toggled simultaneously, the shift register is definitely a synchronous circuit. Note that the asynchronous clear inputs on each flip-flop have been connected together to form a reset line. Application of a low or binary 0 level to this line causes the shift register to be reset. This shift register can also be preset by using any of the techniques described earlier for presetting binary counters. Data applied to the input will be shifted to the right through the flip-flops. Each clock or shift pulse will cause the data at the input and that stored in the flip-flops to be shifted one bit position to the right.

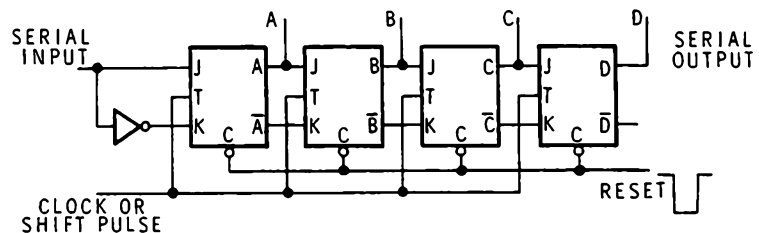


Figure 7-30
Four bit shift register
made with JK flip-flops.

The waveforms in Figure 7-31 illustrate how a serial data word is loaded into the shift register of Figure 7-30. As the waveforms show, the binary number 0101 in serial form occurs in synchronization with the input clock or shift pulses. In observing the waveforms in Figure 7-31, keep in mind that time moves from left to right. This means that the clock pulses on the right occur after those on the left. In the same way, the state of the serial input shown on the left occurs prior to the states to the right. With this in mind, let's see how the circuit operates.

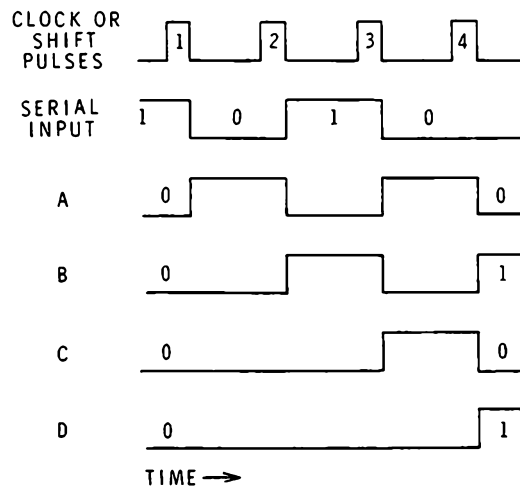


Figure 7-31
Waveforms illustrating
how the serial binary number
0101 is loaded into a shift register.

Note that the shift register is originally reset. The A, B, C, and D outputs of the flip-flops therefore are binary 0 as indicated in the waveforms. Prior to the application of the first shift pulse, the serial input state is binary 1. This represents the first bit of the binary word to be entered. On the trailing edge of the first clock pulse, the binary 1 will be loaded into the A flip-flop. The JK inputs of the A flip-flop are such that when the clock pulse occurs the flip-flop will become set. This first shift pulse is also applied to all other flip-flops. The state stored in the A flip-flop will be transferred to the B flip-flop. The states stored in the B and C flip-flops will be transferred to the C and D flip-flops respectively. Since all flip-flop states are initially zero, naturally no state changes in the B, C, or D flip-flops will take place when the first clock pulse occurs.

After the first clock pulse the A flip-flop is set while the B, C, and D flip-flops are still reset. The first clock pulse also causes the serial input word to change. The clock or shift pulses are generally common to all other circuits in the system and therefore any serial data available in the system will generally be synchronized to the clock.

The input to the A flip-flop is now binary 0. When the trailing edge of the second clock pulse occurs, this binary 0 will be written into the A flip-flop. The A flip-flop which was set by the first clock pulse causes the JK inputs to the B flip-flop to be such that it will become set when the second clock pulse occurs. As you can see by the waveforms, when the second clock pulse occurs, the A flip-flop will reset while the B flip-flop will set. The 0 state previously stored in the B flip-flop will be transferred to the C flip-flop, and the C flip-flop state will be shifted to the D flip-flop. At this point the first two bits of the serial data word have been loaded into the shift register.

The serial input is now binary 1 representing the third bit of the serial input word. When the third clock pulse occurs the A flip-flop will set. The zero previously stored in the A flip-flop will be transferred to the B flip-flop. The binary 1 stored in the B flip-flop will now be shifted into the C flip-flop. The D flip-flop remains reset.

The serial input to the A flip-flop is now binary 0. When the trailing edge of the fourth clock pulse occurs, the A flip-flop will reset. The binary 1 stored there previously will be transferred to the B flip-flop. The 0 stored in the B flip-flop will be shifted into the C flip-flop. The binary 1 in the C flip-flop now moves to the D flip-flop. As you can see, after four clock pulses have occurred, the complete four bit binary word 0101 is now shifted into the register as indicated by the states shown in the waveforms. A glance at the flip-flop output waveforms will show the initial binary 1 bit moving to the right with the occurrence of each shift pulse.

While we have illustrated the operation of the shift register with only four bits, naturally as many flip-flops as needed can be cascaded to form longer shift registers. Most shift registers are made up to store a single binary word. In most modern digital systems, shift registers have a number of bits that is some multiple of four.

While shift registers are readily implemented with JK or D type flip-flops, in most applications MSI shift registers are used. MSI shift registers are available in four and eight bit sizes. Here we are going to discuss a typical four bit MSI TTL shift register. You will see how it can perform the basic shift right operation described previously and how it can be connected to shift left or be parallel loaded.

Figure 7-32 shows the logic diagram of a type 74LS95 TTL shift register. It is made up of four flip-flops with the appropriate gating on the JK inputs. A mode control input line controls this input gating. The mode control also operates the clock input selection circuitry. Two clock signals can be used depending upon the state of the mode control.

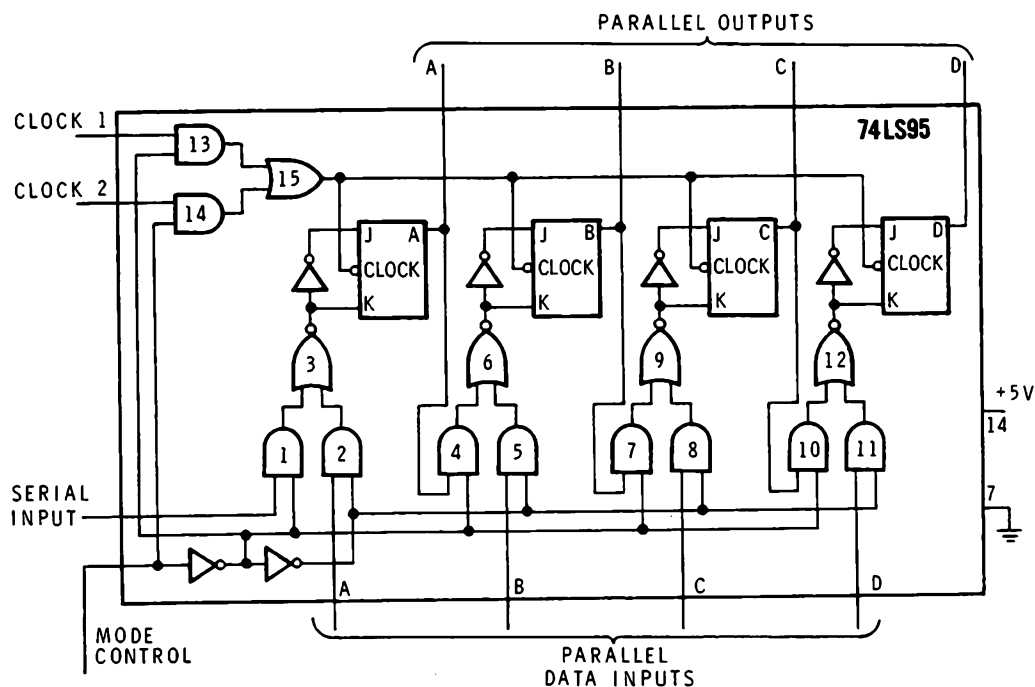


Figure 7-32
Logic diagram of
74LS95 TTL MSI shift register.

When the mode control input is a binary 0, gates 1, 4, 7, and 10 are enabled. This causes the shift register to be set up to perform the basic shift right operation. Serial input is applied to gate 1 and passes through gate 3 to the JK inputs of the flip-flop. The output of the A flip-flop is connected to the inputs of the B flip-flop through gates 4 and 6. In the same way, the outputs of the B and C flip-flops are connected to the inputs of the C and D flip-flops respectively. Also notice that a binary 0 on the mode control input also enables gate 13. This permits clock pulse 1 to pass through gates 13 and 15 to control the flip-flops. In this mode, the shift register performs the standard shift right operation.

When the mode control is placed in the binary 1 state, gates 2, 5, 8, and 11 are enabled. With these gates enabled and gates 1, 4, 7, and 10 inhibited, the parallel data inputs are recognized. Also note that a binary 1 on the mode control input also enables gate 14 so that clock pulse 2 can actuate the flip-flops. When a clock pulse occurs, an external 4-bit parallel word will be loaded into the flip-flops. In this mode, the shift register can be parallel loaded or preset to some desired value.

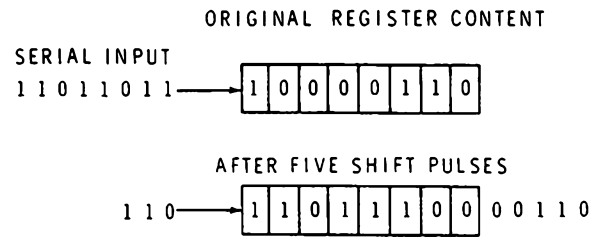
Shift left operations can be performed with the mode control in the binary 1 position if the parallel data input lines are connected to the appropriate flip-flop outputs. To perform a left shift, the D flip-flop output is connected to the C data input, the C output is connected to the B data input, and the B flip-flop output is connected to the A data input. The D data input is used as the serial input line for external data. When clock pulses are applied to gate 14, data will be shifted left from D to C, C to B, and B to A. External serial data is shifted into D. With this connection, the mode control input acts as a shift right, shift left control line.

Self Test Review

27. An 8 bit shift register contains the number 10000110. The serial number 11011011 is applied to the input. After 5 shift pulses, what is the number in the shift register? (Assume shift right operation.)
28. Shift registers can be made up of _____ or _____ type flip-flops.
29. Most shift registers in modern digital applications are of the _____ type.
30. Which of the following operations are *not* typical of the 74LS95 MSI shift register?
 - a. shift right
 - b. shift left
 - c. serial in
 - d. serial out
 - e. reset
 - f. parallel load
31. How many shift pulses are required to serially load a 16 bit word into a 16 flip-flop shift register? _____
32. How could you reset an 8 bit serial in-serial out shift register?

Answers

27.



28. JK, D

29. MSI

30. e. reset The 74LS95 shift register does not provide a separate clear input line.

31. 16

32. Shift in eight binary 0s.

SHIFT REGISTER APPLICATIONS

The shift register is basically a storage element for a binary word. Data can be conveniently shifted into and out of the register in serial form. Despite its simplicity, the shift register has many applications. In this section we are going to look at some of the more popular uses for shift registers.

Serial to Parallel Conversion. One of the most common applications of a shift register is serial to parallel or parallel to serial data conversions. There are many occasions in digital systems where it is necessary to convert an existing parallel word into a serial pulse train. The shift register can readily perform both of these operations.

Figure 7-33 shows how a shift register is used in serial to parallel and parallel to serial data conversions. In Figure 7-33A the shift register is shown being loaded by a parallel input. The number 1101 is preset into the shift register. Then four clock pulses are applied so that the data is shifted out serially. In Figure 7-33B, the shift register is used for serial to parallel conversion. Here the serial input number 1001 is shifted into the shift register by four clock pulses. Once the data is in the register, the outputs of the individual flip-flops may be monitored simultaneously to obtain the parallel output data.

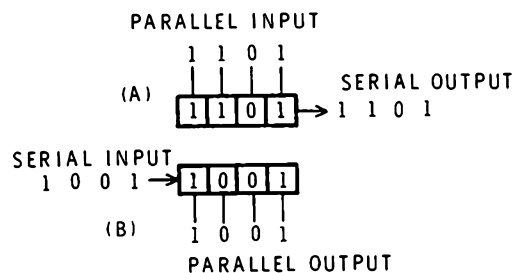


Figure 7-33
Parallel to serial and serial to parallel
data conversions with a shift register.

Scaling Operations. A shift register can be used to perform arithmetic operations such as multiplication and division. Shifting the binary number stored in a shift register to the left has the effect of multiplying that number by some power of 2. Shifting the data to the right has the effect of dividing the number in the register by some power of 2. Shifting operations are a simple and inexpensive way of performing multiplication and division with binary numbers.

Figure 7-34A shows a shift register containing a binary number. Assuming the binary point is located to the far right, we can then convert the binary number into its decimal value. In the initial condition state this number is 3. Now if we perform a shift left operation and move the binary word one position to the left you can see immediately from Figure 7-34B that a new binary number has been formed. As we shift the data to the left one bit position binary 0s are entered on the right. The new number stored in the register is 6. As you can see, shifting the data one position to the left has the effect of multiplying the original number by 2.

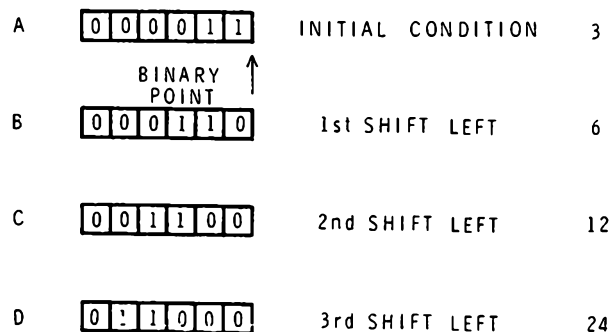


Figure 7-34
Multiplication by powers
of 2 by shifting left.

If we perform another shift left operation the number in the register becomes as shown in Figure 7-34C. Converting it to decimal we see that it is 12. The additional shift left operation has again multiplied the number in the register by 2. Two shift left operations have caused the initial number to be multiplied by four.

A third shift left operation will further verify this effect. The number shown in Figure 7-34D is now 24. Shifting the word one additional position to the left has multiplied the number previously by 2. With three shift left operations the initial number in the register has been multiplied by eight. As you can see then, the factor by which the number in the register is multiplied is some power of 2. The multiplying factor is 2^N where N is the number of shift left operations that take place. With three shift left operations as in Figure 7-34, the original number is multiplied by $2^3 = 8$. The important thing to remember about this operation is the shift register must be large enough to accommodate the largest number expected by multiplication. In addition, note that binary 0s are shifted into the right most position as the data is moved to the left.

Division by some power of 2 is accomplished by shifting right. This is illustrated in Figure 7-35. Here the number initially stored in the register is 20.0. Note in this 6 bit register the binary point has been specified as being between the right most bit position and the next most significant bit.

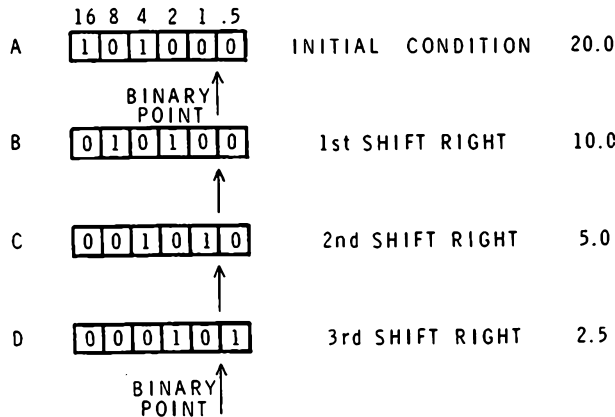


Figure 7-35
Division by powers
of 2 by shifting right.

Applying one clock pulse causes the data in the register to be shifted one position to the right. Evaluating the new decimal value of this number we find that it is 10.0. The number initially stored in the register has been divided by 2. Applying another shift pulse causes the data to be moved one more position to the right. Evaluating this number we find that it is 5.0. Again the number has been divided by 2 while the overall division accomplished by two shift pulses is 4. Performing a third shift right operation moves the data again one bit to the right. Evaluating the new number we find it to be 2.5. Again the data has been divided by 2.

The value by which the number in the register is divided is some power of 2. The divide ratio is 2^N where N is the number of shift right operations. Here the original number 20 was divided by $2^N = 2^3 = 8$ or $20 \div 8 = 2.5$. Again an important consideration is that the register be large enough to accommodate the numbers resulting from the scaling operations by shifting. If the register is not large enough, data will be shifted out of the register and lost thereby making the arithmetic operation incorrect.

Shift Register Memory. Shift registers, because they store binary data, are often used for temporary memories in digital equipment. Such a shift register memory is usually capable of storing at least one binary word. Many such memories are made long enough to store many binary words. In such an application, there are two operations that the memory must perform. First, it must be able to accept data and then store it. In other words, we must be able to write new data into the memory. Second, we must be able to retrieve that data or read it out upon command. One of the requirements of the memory is that when we do read the data out that it will not be lost. A shift register can accomplish both of these operations by providing external logic circuitry as shown in Figure 7-36. Here an eight bit shift register is used to store a single binary word. The external control gates are used to select a read or write operation. To store or write data into the memory, the write/recirculate line is set to the binary 1 state. This causes gate 1 to be enabled. Serial data applied to the other input of gate 1 is passed on into the shift register. Once data is stored in the memory, the write/recirculate control line is set to the binary 0 condition. This inhibits gate 1 and prevents other data appearing at the input from being recognized by the shift register. Instead gate 2 is now enabled. Note that the shift register output is connected to gate 2. As shift pulses are applied, the data in the register is shifted out serially and may be used by some external circuit. As the data is being shifted out it is also being shifted back into the input of the shift register through gates 2 and 3. In other words, we are recirculating the data in the register. The read out operation is accomplished and at the same time the data is restored. When we wish to write a new word into the shift register, the write/recirculate line is again made binary 1 and 8 shift pulses are applied.

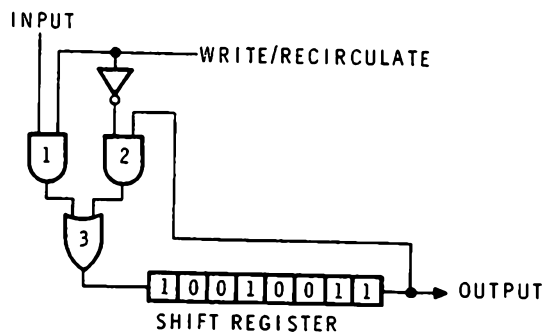


Figure 7-36
Shift register memory.

Sequencer/Ring Counter. Another popular application of the shift register is a sequencer or ring counter. Many logic circuits require a sequence of equally spaced timing pulses for initiating a series of operations. A properly connected shift register can be used to serve this purpose.

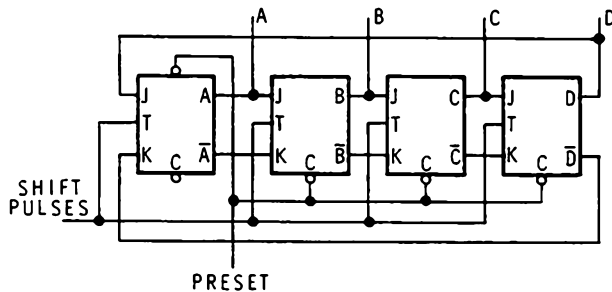
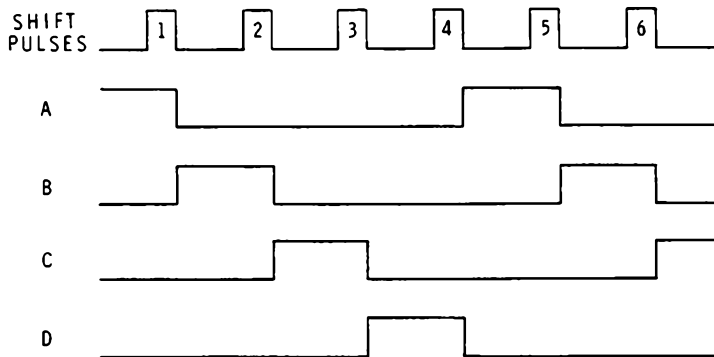


Figure 7-37
A shift register connected as a sequencer or ring counter.

A shift register connected as a ring counter is shown in Figure 7-37. This is the standard shift register circuit we discussed earlier. However, note that the outputs of the shift register from the D flip-flop are connected back to the JK inputs of the A flip-flop. This provides feedback that causes the shift register to continue to rotate or sequence the data in the register. The asynchronous set and clear inputs of the flip-flop are used to preset a single bit in the shift register. A binary 0 level applied to the preset line causes the A flip-flop to become set and the other three flip-flops to become reset. As shift pulses are applied, the binary 1 in the A flip-flop is shifted to the B, C, and D flip-flops, and then back around to the A flip-flop. This sequence repeats as long as the shift pulses are applied. Since the one bit initially programmed into the shift register is continually recirculated, the name ring counter definitely applies. The waveforms and the state table in Figure 7-38 show the operation of the four bit shift register as a ring counter.



STATE	A	B	C	D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

RECYCLE

Figure 7-38
Waveforms for a Four-bit shift register ring counter. State table for ring counter.

A close look at the waveforms for the four bit ring counter in Figure 7-38 shows that the output at any flip-flop has a frequency one fourth that of the shift pulse frequency. In other words, the shift register connected as a ring counter produces frequency division by four since four shift input pulses occur for each flip-flop output pulse. As you can see then the shift register when connected as a ring counter can be used as a frequency divider. By presetting one of the flip-flops in the shift register, frequency division by any integer value can be accomplished by simply using as many flip-flops as needed. For example, to divide by seven, seven flip-flops would be required in the ring counter.

One of the disadvantages of the ring counter circuit shown in Figure 7-37 is that it must be initially preset in order for it to function properly. When power is initially applied to this circuit, the flip-flops in the register can come up in any state. If any random state is allowed in the shift register, the operation previously described will not occur. The preset operation must take place to initially load one of the flip-flops with a binary 1 and the others with binary 0. This disadvantage can be overcome by using a self correcting circuit as shown in Figure 7-39. Here the NAND gate monitors the outputs of flip-flops A, B, and C. The output of the NAND gate and its complement are connected to the JK inputs of the A flip-flop. With this circuit arrangement, any of the sixteen possible states can occur in the shift register and the shift register will automatically correct itself to where only one of the flip-flops is set and the others are reset. Regardless of the initial states of the flip-flops, after a maximum of three shift pulses, the contents of the shift register will automatically be corrected so that only a single flip-flop is set. From that point on the shift register will simply recirculate the single one bit stored there.

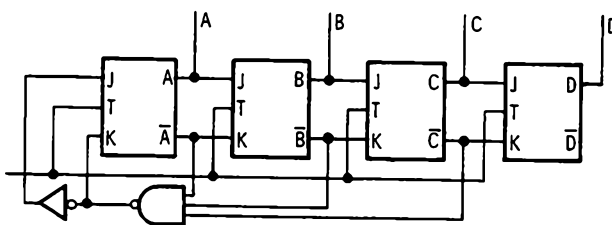


Figure 7-39
Self correcting shift register ring counter.

To use the ring counter described here as a sequencer, the output pulses from the flip-flops are simply connected to the logic circuits whose sequence is to be controlled. Since the shift pulses are normally derived from a fixed frequency clock, the timing interval of the shift registers is precise and thereby permits very exacting control of the external logic circuits. In addition, it is not necessary to use all of the pulses derived by the shift register. Only those required by the circuit need be used. Keep in mind that as more circuits must be controlled or sequenced, additional flip-flops can be added to the shift register to produce them.

Counters. When connected as a ring counter, shift registers can also be used as a counter. In some special applications they may replace binary counters for certain operations. The four bit ring counter circuit described previously has four distinct states and these states repeat or recycle as the clock pulses are applied.

A popular type of shift register counter is the Johnson counter shown in Figure 7-40. While any number of flip-flops may be cascaded to form a Johnson counter, a five bit circuit is often used. Note that like in the ring counter, the output of the last flip-flop is connected back to the inputs of the first flip-flop in order to recirculate the data. However, note that in this case the normal output of the E flip-flop is connected to the K input and the complement output is connected to the J input. Because of this connection the Johnson counter is often referred to as a twisted ring counter or switch tail counter. With this arrangement, the counter or shift register will have $2N$ different states where N is the number of flip-flops in the shift register. The five bit Johnson counter shown in Figure 7-40 therefore, will have ten discrete states.

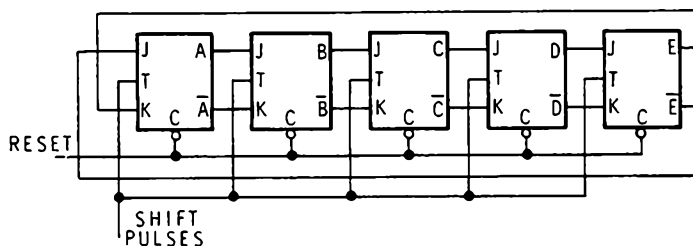


Figure 7-40
Shift register connected
as a Johnson counter.

Like the ring counter shift register discussed previously, it is necessary to initialize the counter after power is applied in order to have the counter operate properly. A self correcting circuit similar to that shown in Figure 7-39 can be used to initialize the circuit. Otherwise initialization can be accomplished by simply resetting all of the flip-flops to zero. When shift pulses are applied, the binary state sequences shown in the table of Figure 7-41 are generated. Note the ten individual states. The counter recycles every tenth input pulse. Since the Johnson counter has ten individual states, it is often used as a divide by 10 frequency divider. Other counting operations can often be implemented with this type of counter. However, because of the nonweighted codes generated by the Johnson counter and the ring counter shift register, many counting applications are difficult and inconvenient to implement.

STATE	A	B	C	D	E
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	1	1	1	1	0
5	1	1	1	1	1
6	0	1	1	1	1
7	0	0	1	1	1
8	0	0	0	1	1
9	0	0	0	0	1

RECYCLE

Figure 7-41
State table for Johnson counter.

Self Test Review

33. A binary number is shifted 5 positions to the left. The number therefore has been
 - a. multiplied by 5
 - b. divided by 5
 - c. multiplied by 32
 - d. divided by 32
34. A binary number must be divided by 128. How many positions must the number be shifted (and in what direction) to achieve this?
35. How many flip-flops must be used in a ring counter shift register to perform frequency division by 12?
 - a. 4
 - b. 6
 - c. 12
 - d. 24
36. How many states does a Johnson counter with 8 flip-flops have?
 - a. 8
 - b. 16
 - c. 32
 - d. 256
37. Shift registers can operate in both serial and parallel modes. List all four possible combinations of these modes.
 - a. _____
 - b. _____
 - c. _____
 - d. _____
38. In our discussion of using a shift register to multiply and divide by shifting, we assumed that the right most bit was the LSB. What happens to our rules about multiplication and division by shifting if the left most bit is made the LSB?

Answers

- 33. c. multiplied by 32.
- 34. 7 positions to the right.
- 35. c. 12
- 36. b. 16
- 37.
 - a. Serial In-Serial Out (SI-SO)
 - b. Parallel In-Parallel Out (PI-PO)
 - c. Serial In-Parallel Out (SI-PO)
 - d. Parallel In-Serial Out (PI-SO)
- 38. The rules are reversed. A right shift now becomes a multiply and a left shift becomes a divide.

MOS SHIFT REGISTERS

In applications requiring shift registers with a limited bit capacity, bipolar integrated circuits such as TTL or ECL are used. CMOS shift registers are also available. Such registers are generally used for storing a single binary word. This word may be as small as four bits but could be as long as 32 bits in some applications. Where more data storage is needed, additional flip-flops or MSI shift registers can be cascaded to form memories which can be used for storing many binary words. When implemented with standard bipolar and CMOS integrated circuits, such shift register memories become very large and expensive. In these applications MOS shift registers can be of value.

MOS integrated circuit shift registers using P or N channel enhancement mode MOSFETs contain many storage elements. Because of the very high component density and very low power dissipation of the MOS structure, very large shift registers can be made on a very tiny silicon chip. MOS shift registers with thousands of storage elements are available for memory applications. Such shift registers are commonly used to store many binary words in a serial format. For example, to store 128 eight bit binary words we would need an $8 \times 128 = 1024$ bit shift register.

MOS shift registers this large are very practical and are commonly used for temporary data storage and for delay operations. Any application requiring the temporary storage of a large volume of binary data can use MOS shift registers. In addition, most MOS LSI shift registers are of the serial in/serial out type. The parallel loading and readout of data is not generally performed with MOS shift registers. These MSI and LSI circuits provide a very economical memory source.

There are two basic types of MOS shift registers: static and dynamic. A static shift register is one in which the clock may be stopped without loss of data. This is the type of shift register that we have discussed in the previous sections. Clock signals are applied to shift data into or out of the register. When the clock pulses are stopped, the data in the shift register is retained in the storage elements. Data is not lost if the clock is stopped.

In another type of MOS shift register the data will be lost if the clock is stopped. This type of shift register is known as the dynamic type.

Because of the characteristics of the storage element used in a dynamic register, the clock pulses must run continuously if data is to be retained. Data must be continuously recirculated or refreshed in order to prevent its loss. Naturally, the static shift register is generally more desirable from a standpoint of operation and convenience. However, static MOS shift registers are generally more complex and consume much more power.

Dynamic shift registers can be made smaller and more simply, can operate at higher speeds and have far lower power dissipations. Such trade offs must be considered when designing with MOS shift registers. Most MOS shift registers are fully compatible with TTL and CMOS circuits. No special interfacing is required.

Dynamic MOS Shift Register. The basic storage element in a MOS shift register, whether it is dynamic or static, is the capacitance that exists between the gate and the channel of the MOSFET transistors used. While this capacitance is very small (on the order of several tenths of a picofarad), the high impedance nature of the MOSFET permits a charge voltage to be placed on this capacitance and retained for a relatively long period of time. The impedance between the gate and the source of an enhancement mode MOSFET is on the order of 10^{15} ohms or greater. Such a high impedance is virtually an open circuit and has a minimum effect on the gate capacitance. If we apply a voltage between the gate and source of a MOSFET, the gate capacitance will charge and remain there until it leaks off through the very high impedance between the source and gate. In high quality MOSFETs, this discharge time can be as long as one millisecond.

The storage element circuit used in a MOS shift register is a MOSFET inverter. The input capacitance of the inverter transistor stores the data. Figure 7-42 shows how two MOSFET inverters (I1 and I2) are combined with MOSFET transmission gates (Q_1 and Q_2) to form a

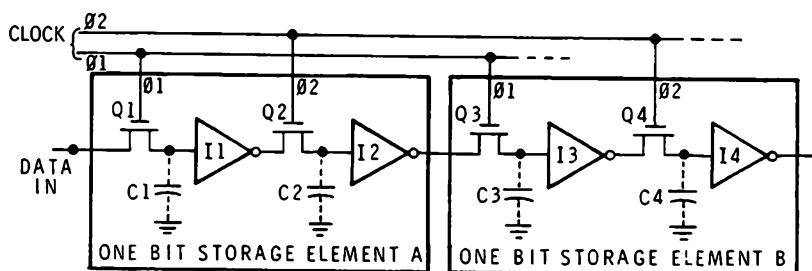


Figure 7-42
Dynamic MOS shift register.

one bit storage element. The input data is applied to inverter I1 through transmission gate Q1. MOSFET Q1 is simply used as an on/off switch to connect the input to capacitor C1 and disconnect it. The output of inverter I1 is connected to the input of inverter I2 through transmission gate Q2 which again is used as a simple on/off switch. The switching of the transmission gate transistors is controlled by two clock signals designated phase 1 ($\phi 1$) and phase 2 ($\phi 2$). These two phase clock signals are illustrated in Figure 7-43. Note that when $\phi 1$ is on $\phi 2$ is off and vice-versa.

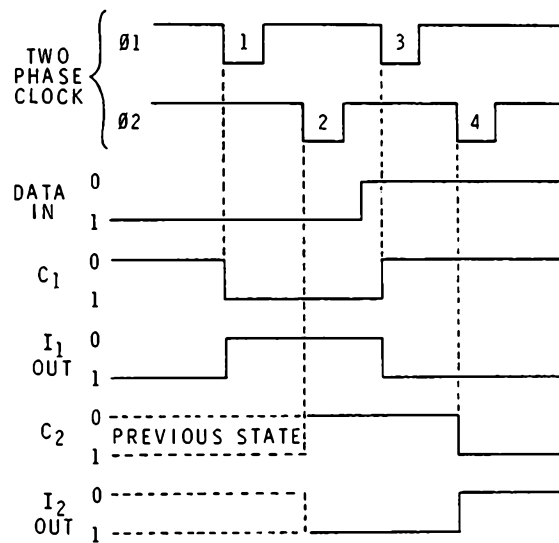


Figure 7-43
Clock and circuit waveforms
for dynamic MOS shift register.

For our discussion here let's assume the use of P channel MOS circuits where a binary 0 is equal to zero volts or ground and a binary 1 is some negative voltage level.

The data to be stored (written) into storage element A is applied to the data input line. Assume that we apply a binary 1 input which is some negative voltage level. When clock pulse $\phi 1$ occurs, transmission gate Q1 will conduct. This will cause capacitor C1 to charge to the input voltage. Applying a binary 1 input voltage to inverter I1 causes a binary 0 level to appear at its output. After the occurrence of the $\phi 1$ clock pulse, capacitor C1 retains the charge and acts as the input voltage source for inverter I1.

The $\phi 2$ clock pulse occurs next. This causes transmission gate Q2 to conduct. The state of the output of I1 is, therefore, transferred to capacitor C2. This is a binary 0, so capacitor C2 has zero charge. The input to inverter I2 is a binary 0. The output of I2, therefore, is a binary 1. After one $\phi 1$ clock pulse and one $\phi 2$ clock pulse, the binary 1 that was at the input to storage element A appears at the output of storage element A. On the next cycle of $\phi 1$ and $\phi 2$ clock pulses, this binary 1 value will be transferred to the next storage element (B) of the shift register. Any new data appearing at the input of the first storage element will be shifted in at this time. The waveforms in Figure 7-43 illustrate the storage of a binary 1 in element A and its transfer to element B as a binary 0 is entered.

The inverters in Figure 7-42 can be any one of several different types of MOS logic inverters. Figure 7-44 shows the two types most commonly used. In Figure 7-44A, a static inverter is used. Here Q2 is the inverter element while Q1 is a MOSFET biased into conduction to act as a load resistance. This type of device dissipates power because Q1 is continuously conducting. In long MOS shift registers this power dissipation is additive and can produce a significant amount of heat.

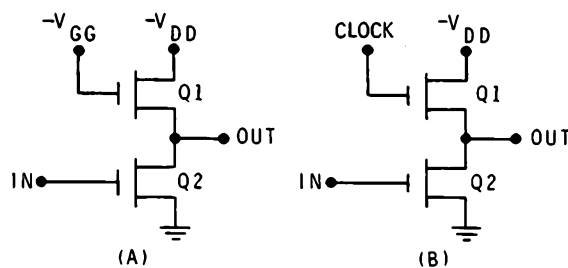


Figure 7-44
MOS inverters (A) static and (B) clocked.

Another type of inverter shown in Figure 7-44B uses a clocked load device. Here, Q2 is the inverter element while Q1 is the load element. Q1, however, does not conduct, except during the time a clock pulse is applied. When such an inverter is used in the dynamic shift register circuit at Figure 7-42, the load element is clocked during $\phi 1$ or $\phi 2$ along with the associated output transmission gate. For example, the load element in I1 would be clocked at $\phi 2$ time while the load element in I2 would be clocked at $\phi 1$ time. This arrangement greatly reduces the power dissipation of the device.

In order to keep the data from being lost during the shifting process, the clock must run continuously and the data must be continually recirculated from output to input. Write/recirculate logic at the input of the shift register is used to select the mode of operation. Should the clock pulses stop, the data stored as charges on the capacitances in the circuits will leak off and be lost. The loss of data can occur in only several hundred microseconds depending upon the circuitry used. For that reason the minimum clock rate is approximately 5 kHz for most typical dynamic MOS shift registers. Dynamic shift registers with minimum clock rates in the 100 Hz range are available.

Static MOS Shift Registers. There are some applications where it is desirable to stop the clock in a digital system. For such applications, static MOS shift registers can be used. Such shift registers employ storage elements that retain the data even after the clock has stopped. In addition, it is not necessary to continually recirculate the data in order to retain it.

A typical storage element for a static MOS shift register is shown in Figure 7-45. It also uses the gate capacity of a MOSFET inverter for temporary data storage. The storage element uses two such inverters. One inverter is transistor Q3 with its load device Q2. The other inverter is Q7 with its load device Q6. These two inverters are cross coupled through transmission gates Q4 and Q5. When Q4 and Q5 conduct, a latch type flip-flop is formed. Naturally, a latch will retain data even when the clocking signals are removed as long as Q4 and Q5 conduct. Transistor Q1 is used as a transmission gate to load data into the storage element. To explain the operation of the circuit, assume that P channel devices and negative logic are used.

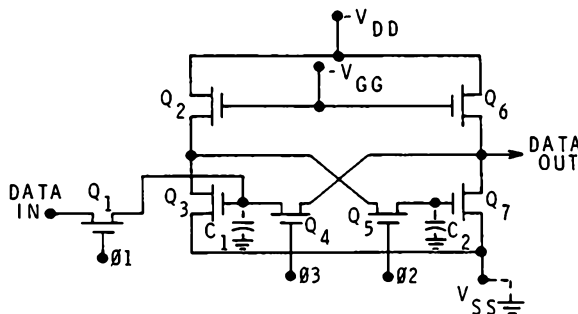


Figure 7-45
One bit MOS storage element
for a static MOS shift register.

The proper operation of this shift register requires a three phase clock signal. These clock signals are shown in Figure 7-46. The ϕ_3 clock is a delayed replica of the ϕ_2 clock signal. In some static MOS shift registers, the ϕ_3 and sometimes the ϕ_2 and ϕ_3 clock pulses are generated on the chip. Therefore, the circuit requires only a single or double phase external clock for proper operation.

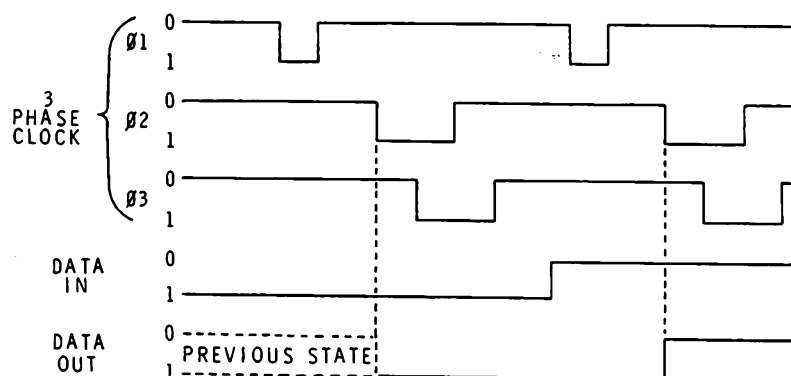


Figure 7-46
Waveforms for static MOS shift register.

To load data into the circuit, the desired bit is applied to the data input line. When the ϕ_1 clock occurs, transmission gate Q1 conducts. This causes the gate capacitance of Q3 (C1) to charge to the proper state. If a binary 1 is applied to the data input, C1 will assume a negative charge. This negative charge is applied to inverter Q3. Q3 conducts and its drain goes low representing a binary 0.

The ϕ_2 clock occurs next and Q5 conducts, thereby, transferring the state of Q3 to capacitor C2. In our example, this is a binary 0. The output of inverter Q7 then is a negative level binary 1. The ϕ_3 clock signal occurs and Q4 conducts. This applies the negative signal back to the gate of Q3 to keep it on. At this time, the data is latched. The shift register will remain in this state until the state of the input has changed and the next clocking cycle has been completed.

Self Test Review

39. MOS shift registers are which of the following type?
- a. SI-SO
 - b. SI-PO
 - c. PI-PO
 - d. PI-SO
40. The two types of MOS shift registers are _____ and _____.
41. Static MOS shift registers consume
- a. more
 - b. less
- power than a dynamic register.
42. MOS shift registers are used mainly
- a. for storing a single binary word.
 - b. parallel to serial data conversions.
 - c. for multiplying and dividing operations.
 - d. as a memory for storing many binary words.
43. The main storage element in an MOS register is the _____ of a MOSFET inverter.
44. What two requirements are necessary to prevent a data loss in a dynamic MOS shift register?
- a. _____
 - b. _____

Answers

- 39. a. SI-SO
- 40. static, dynamic
- 41. a. more
- 42. d. as a memory for storing many binary words.
- 43. gate capacitance
- 44. a. continuously running clock
 - b. data recirculation

CLOCKS AND ONE-SHOTS

Most sequential logic circuits are driven by a clock. The clock is a periodic signal that causes logic circuits to be stepped from one state to the next. The clock steps the sequential circuits through their normal operating states so that they perform the function for which they were designed.

The clock signal is generated by a circuit known as a clock oscillator. Such an oscillator generates rectangular output pulses with a specific frequency, duty cycle and amplitude. The most commonly used clock oscillator is some form of astable multivibrator. Such circuits can be constructed with discrete components or with logic gates.

Another circuit widely used to implement sequential logic operations is the one shot. The one shot or monostable multivibrator produces a fixed duration output pulse each time it receives an input trigger pulse. The duration of the pulse is usually controlled by external components. By cascading one shot circuits, a wide variety of sequential circuits can be implemented.

Clock Oscillator Circuits

Practically all digital clock oscillator circuits use some form of astable multivibrator circuit for generating a periodic pulse waveform. Such a circuit has two unstable states, and the circuit switches repeatedly between these two states. Both discrete component and integrated circuit clocks are used in digital equipment.

Discrete Component Circuits. The most commonly used clock oscillator is the astable multivibrator circuit shown in Figure 7-47. It consists of two transistor inverters Q_1 and Q_2 with the output of one connected to the input of the other. Resistors R_2 and R_3 are used to bias the transistors into saturation. Capacitors C_1 and C_2 couple the output of one inverter to the input of the other. In normal operation, one transistor is conducting while the other is cut off. The frequency of oscillation is determined by the values of R_2 , R_3 , C_1 , and C_2 .

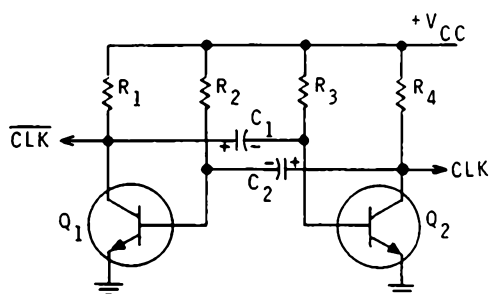


Figure 7-47

Astable multivibrator clock oscillator.

Assume that Q2 is conducting and Q1 is cut off. Capacitor C1 then charges through the emitter-base junction of Q2 and R1 to the supply voltage $+V_{CC}$. Capacitor C2 which was previously charged to the supply voltage with the polarity shown keeps Q1 cut off as it discharges through resistor R2. As soon as it discharges to zero it begins to charge in the opposite direction. When the charge on C2 reaches about 0.7 volt, Q1 conducts. As soon as Q1 conducts, it effectively connects the positive side of C1 to ground. This puts a negative voltage between the base and the emitter of Q2 causing it to switch off quickly. C1 then discharges through R3. At this time C2 is recharged through the emitter-base junction of Q1 and R4. As soon as the charge on C1 has reached zero, it will begin to charge to the supply voltage. However, as soon as the voltage is high enough, Q2 again conducts and the state of the circuit reverses. This cycle continues at a rate determined by the discharge time of C1 and C2 which in turn depends upon the values of R2 and R3. R2 and R3 are generally selected in order to ensure saturation of Q1 and Q2. Capacitors C1 and C2 are then chosen to produce the desired operating frequency with the given base resistors. The frequency of oscillation (f) is approximately equal to:

$$f = \frac{1}{1.4 RC}$$

This formula assumes the $R = R_2 = R_3$ and $C = C_1 = C_2$. With this arrangement the circuit will produce a 50 percent duty cycle output square wave. Unequal values of capacitors can be used to produce a duty cycle more or less than 50 percent.

Figure 7-48 shows the circuit outputs. The outputs are taken from the collectors of the two transistors and they are complementary. The outputs switch between the supply voltage $+V_{CC}$ and the $V_{CE}(\text{sat})$ of each transistor. This clock oscillator circuit can drive most standard logic families such as TTL and CMOS directly. For other types of logic, interface circuitry may be required between the clock oscillator and the logic circuitry.

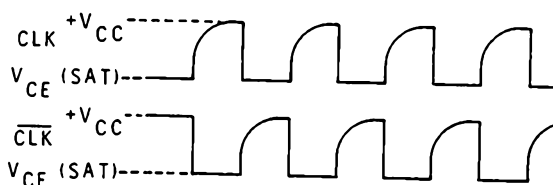


Figure 7-48
Waveforms for the astable multivibrator.

IC Clock Circuits. The astable multivibrator of Figure 7-47 can also be implemented by using integrated circuit gates or inverters. Figure 7-49A shows the astable circuit implemented with TTL inverter circuits. The operation of this circuit is practically identical to the circuit in Figure 7-47. The frequency of oscillation is a function of the values of resistance and capacitance in the circuit. The resistors provide a charge path for the capacitors and are also used to provide bias to the inverter circuits. The frequency of oscillation of this circuit is approximately equal to:

$$f = \frac{1}{2 RC}$$

Where f is in kHz, R is in k ohms and C is in microfarads. The output of the circuit will be 50 percent duty cycle square wave. Inverter 3 is used to buffer the circuit output and to isolate the load from the frequency determining components.

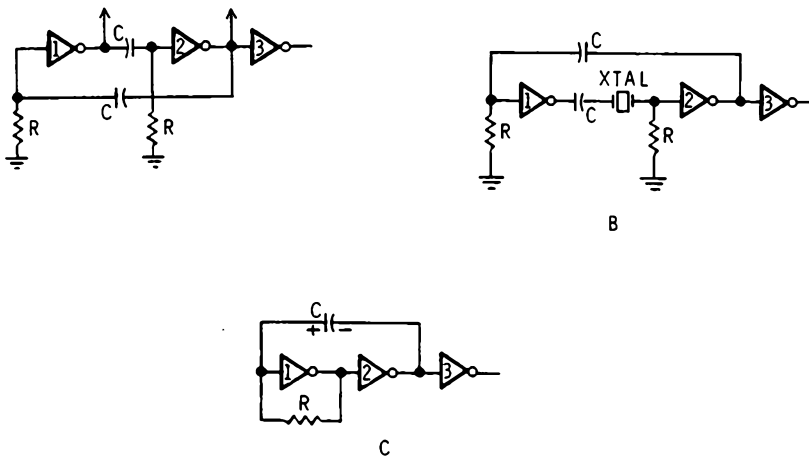


Figure 7-49

Astable multivibrators made with TTL inverters.

- (A) Conventional astable,
- (B) crystal controlled astable,
- (C) simplified astable.

The basic IC astable multivibrator can be modified as shown in Figure 7-49B to include a frequency determining crystal. The values of R and C are selected to perform oscillation near the desired frequency. The circuit will oscillate at the crystal frequency. This circuit is desirable when the clock frequency must be very accurate and remain stable. Again inverter 3 is used to buffer the output and isolate the load from the frequency determining components.

Figure 7-49C shows another version of the basic astable multivibrator. This circuit uses only a single RC network. Resistor R is used to bias inverter 1 close to the linear region. In this circuit, the value of R is very critical and should be somewhere in the 150 to 220 ohm range when standard TTL inverters or OR gates are used.

The operation of this astable multivibrator is somewhat different from the discrete component circuit and its integrated circuit counterparts discussed earlier. This is how it operates.

Refer to Figure 7-49C. Assume the output of inverter 2 goes low. This low will be coupled through capacitor C to the input of inverter 1, therefore, the output of inverter 1 will go high. The high input to inverter 2 ensures that its output remains low. At this time capacitor C charges through R to the output voltage of inverter 1. When the voltage to the input of inverter 1 reaches approximately 1.5 volts, the output of inverter 1 will go low forcing the output of inverter 2 high. The high output from inverter 2 plus the charge on capacitor C ensures a high level to the input of inverter 1 keeping its output low. Capacitor C now begins to discharge through R. As soon as the charge on C becomes low enough, the output of inverter 1 will switch high causing the output of inverter 2 to go low. The cycle will then repeat itself. The period (p) of oscillation of this circuit is approximately 3 RC. As in the other circuits, inverter number 3 is used to isolate the load from the frequency determining components and to ensure a clean square wave output.

$$p = 3RC$$

$$f = \frac{1}{3RC}$$

Two-Phase Clocks. The circuits we have discussed generate a single-phase clock. For most MOS integrated circuits, a two-phase clock is required. Bipolar logic circuits and CMOS usually operate from a single-phase clock. There are several different methods of generating two-phase clock signals, but one of the most commonly used methods is shown in Figure 7-50. Two TTL JK flip-flops are connected to form

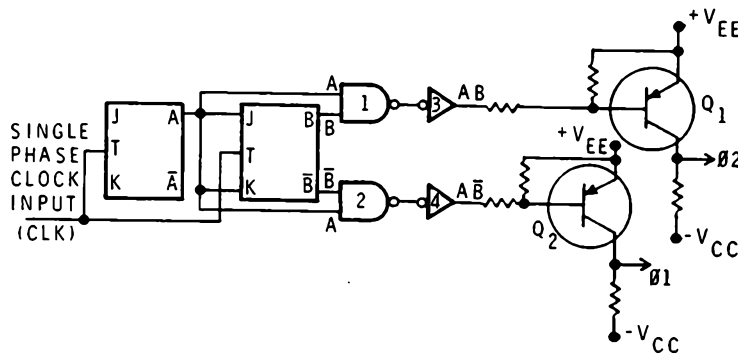


Figure 7-50
Two-phase clock generator.

a synchronous 2-bit binary counter. The count sequence is 00, 01, 10, and 11. Gates 1 and 2 are used to detect the AB and $A\bar{B}$ states of the counter. These gates are used to drive transistors Q_1 and Q_2 which form the interface circuitry for developing the proper logic levels for use with PMOS integrated circuits. The phase 1 (ϕ_1) and phase 2 (ϕ_2) clock signals switch between $+V_{EE}$ and $-V_{CC}$. These levels are typically $+5$ and -5 volts. The waveforms for the two-phase clock circuitry are shown in Figure 7-51.

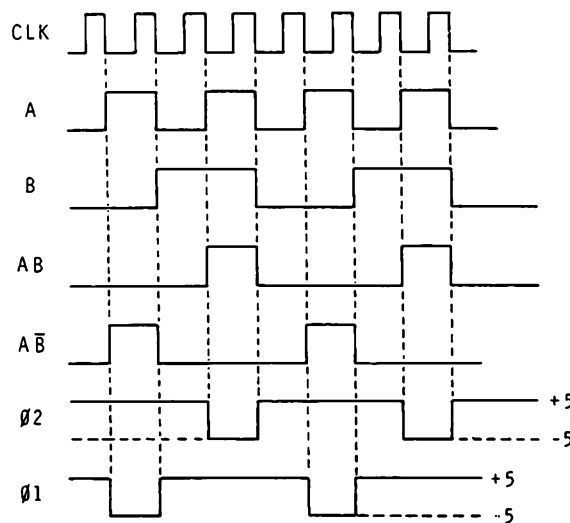


Figure 7-51
Waveforms for two-phase clock generators.

When both flip-flops are set, the output of inverter 3 goes high. Q1 cuts off and the $\phi 2$ output becomes $-V_{CC}$. When the output of inverter 3 goes low, Q1 conducts and the output becomes $+V_{EE}$. The operation of Q2 is similar. When flip-flop A is set and B is reset, the output of inverter 4 goes high. Q2 cuts off and the $\phi 1$ output goes to $-V_{CC}$. When the output of inverter 4 is low, Q2 conducts and the $\phi 1$ output is $+V_{EE}$. The waveforms in Figure 7-51 show this sequence.

One Shot Multivibrators

The one shot or monostable multivibrator is a circuit that generates a rectangular output pulse of a specific time duration each time it receives an input trigger pulse. The output pulse duration is usually adjustable by varying the value of external circuit components. By cascading these circuits, a variety of sequential logic operations can be implemented.

Discrete Component One Shot Circuit. Figure 7-52 shows the schematic diagram of a one shot multivibrator. This circuit has two states: a stable state where Q2 conducts and Q1 is cut off and an unstable state where Q1 conducts and Q2 is cut off. The circuit normally rests in its stable state when it is not being triggered. The unstable state is initiated when the circuit receives an input trigger pulse. The one shot then goes into its unstable state and for a period of time depending upon the values of C1 and R2 it generates an output pulse. The circuit then returns to its stable state.

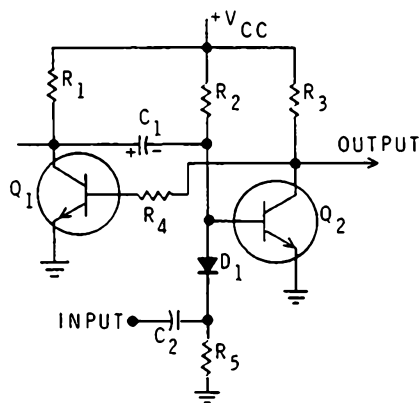


Figure 7-52
One shot multivibrator

In the stable state, resistor R2 forward biases the emitter-base junction of Q2. Q2 saturates and its output is near zero volts or ground. Therefore, the voltage applied to R4 is insufficient to cause Q1 to conduct. Q1 therefore, is cut off and its collector is at $+V_{CC}$. Capacitor C1 charges through the emitter-base junction of Q2 and resistor R1 to a voltage approximately equal to supply voltage $+V_{CC}$.

The circuit will remain in this stable state until it receives an input trigger pulse. To trigger the circuit, an input pulse is applied. The network consisting of C2 and R5 differentiates the input pulse. The sharp positive and negative pulses occurring at the leading and trailing edges of the input waveform are then applied to diode D1. D1 permits only the negative going pulse to be coupled to the base of Q2. The negative going pulse reverse biases the emitter-base junction of Q2. Q2 switches off and its output voltage rises to $+V_{CC}$. This causes Q1 to become forward biased. It receives base current through R3 and R4 from $+V_{CC}$. With Q1 saturated its collector is near zero volts. C1 begins to discharge through resistor R2. The negative voltage from this capacitor at the base of Q2 keeps Q2 cut off. As C1 discharges through R2, its voltage drop becomes smaller. Soon C1 will be completely discharged and will begin to charge to the opposite polarity. When the voltage across it is high enough, it forward biases Q2. As soon as Q2 switches on, the output pulse is terminated. C1 then recharges through the emitter-base junction of Q2 and R1. Figure 7-53 shows the input and output waveforms generated by the one shot circuit.

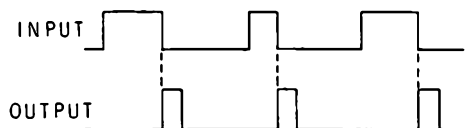


Figure 7-53
Waveforms for the one shot multivibrator.

There are several important facts about the one shot that should be considered in its application. First, the output pulse duration is a function of the values of C1 and R2. The value of R2 is rather critical since it must be low enough in value to ensure the complete saturation of Q2 during normal operation. The value of C1 can be almost any value. The time duration of the output pulse (t) is approximately $t = .69(R_2C_1)$. In most practical monostable multivibrators, the output pulse can be adjusted from nanoseconds to seconds.

Second, the output pulse is initiated on the negative or trailing edge of the input. The positive or leading edge has no effect. See Figure 7-53.

The duty cycle is generally limited to a maximum of approximately 90 percent. A duty cycle greater than approximately 90 percent will generally cause the circuit to operate unreliably. The reason for this is that sufficient time must be provided for the circuit to recover between input trigger pulses. This is the time required for capacitor C1 to completely recharge through the emitter-base junction of Q2 and R1 after a pulse has been generated. This finite charge time for C1 can be reduced by making R1 smaller. However, there is a limitation because of practical circuit considerations. As for minimum duty cycle, there is no practical lower limit. Duty cycles of only a few percent can be achieved with such a multivibrator.

Duty Cycle

Duty cycle is the ratio of the output pulse duration to the total period of the trigger pulse input expressed as a percentage.

$$\text{Duty cycle} = \frac{t}{p} \times 100 \text{ percent}$$

Here t is the pulse duration and p is the period.

As an example, assume the pulse duration is 5 milliseconds and the input frequency is 50 Hz. See Figure 7-54. The input period is $1/50 = .02$ seconds or 20 milliseconds. The duty cycle then is:

$$\text{Duty cycle} = \frac{5}{20} \times 100 = 25 \text{ percent}$$

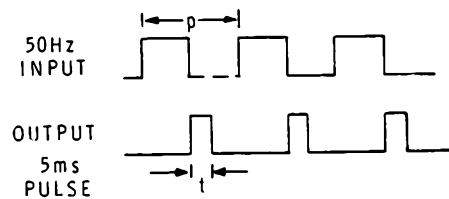


Figure 7-54

Integrated One Shots. Most one shot circuits in use today are in integrated circuit form. Their operation is virtually identical to the discrete component circuit just discussed.

Figure 7-55 shows the logic symbol used to represent these one shots. This circuit has three inputs by which the one shot may be triggered. Inputs A1 and A2 can trigger the one shot if the B input is held high. Inputs A1 and A2 will trigger the one shot on the trailing edge. When A1 or A2 switches from high to low, the one shot will generate an output pulse. Complementary output pulses appear at the Q and \bar{Q} outputs. The duration of the output pulse is a function of the external components C and R. The manufacturer provides guidelines for selecting these values and charts for computing the pulse width for given values of C and R. Generally, the external value of R is limited to approximately 50 k ohms while practically any value of capacitance from 10pf to 100 μ f can be used. Duty cycles as high as 90 percent are possible.

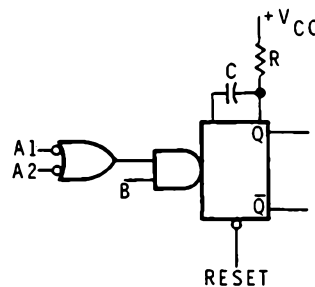


Figure 7-55
Integrated circuit one shot.

Input B can also be used to trigger the one shot if inputs A1 and A2 are not used (held low). The one shot will be triggered when input B switches from low to high. In other words, input B triggers the one shot on the leading edge of the input. This input is used primarily for inhibiting or enabling of inputs A1 and A2. Note also that the one shot has a reset input. This is similar to the asynchronous direct clear input of a JK flip-flop. Bringing this input low automatically terminates the output pulse during a timing period. When the one shot is not triggered, the Q output is binary 0 while \bar{Q} is binary 1. When a trigger pulse is received, the one shot goes into its unstable state where Q is binary 1 and \bar{Q} is binary 0. A reset pulse applied during the timing period will cause the normal output to switch to binary 0, immediately terminating the timing sequence. The waveforms of Figure 7-56 illustrate these operations of the IC one shot.

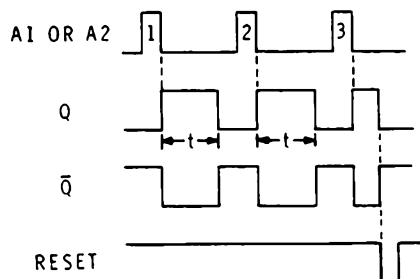


Figure 7-56

Waveforms of an IC one shot.

Input pulses 1 and 2 trigger the circuit into operation on the trailing edge. The output is a pulse whose duration (t) is defined by the values of R and C . Note that the timing interval terminates prior to the application of each new input pulse. On the third input pulse, the one shot is triggered but the timing interval is cut short because of the occurrence of a reset pulse.

Another type of IC one shot circuit available to the digital designer is the retriggerable monostable or negative recovery monostable. Most one shots require a finite period of time in order to recover from a trigger pulse. Once a one shot has been triggered and times out, it will take a short period of time for the capacitor to become recharged through the circuit resistance. It is this recovery time that limits the upper duty cycle of most one shots to approximately 90 percent. The retriggerable monostable eliminates this problem. Its recovery time is practically instantaneous thereby making 100 percent duty cycle outputs almost a possibility. A 100 percent duty cycle represents a constant binary 1 output.

One of the benefits of the retriggerable monostable is its ability to generate very long duration output pulses. By adjusting the external resistor and capacitor values of the one shot to provide an output pulse duration that is longer than the interval between the input trigger pulses, the retriggerable one shot will remain in the triggered state for a substantial period of time. The waveforms in Figure 7-57 illustrate this effect. Initially, the one shot is in its normal stable state. When the trailing edge of input pulse 1 occurs, the monostable is triggered. However, before it can complete its output pulse whose duration is a function of the external component values, input pulse 2 occurs. When its trailing edge occurs the first timing interval is automatically terminated and a new timing interval initiated. This happens quickly so that the output remains high. Note that another input pulse does not occur after input pulse 2 and therefore the one shot is then allowed to time out and generate its normal output pulse width (t).

In addition to generating very long output pulses, the retriggerable one shot can also be used as a missing pulse detector. By making the pulse width of the multivibrator longer than the period of input trigger pulses, the one shot will remain triggered during the sequence of input pulses. If one of the input pulses should disappear or be lost due to a malfunction or noise interference, the one shot will time out. Its output will go low and will therefore indicate the missing pulse.

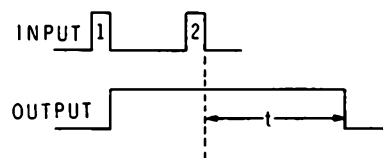


Figure 7-57

Input and output waveforms
of a retriggerable monostable.

One Shot Applications

Because of the flexibility of an integrated circuit one shot, many sequential operations can be quickly and easily implemented. The ability to adjust the output pulse width with external components to a desired value plus the retriggerable and reset features makes the one shot a versatile component. As a result, digital designers find many applications for it. Because of the nature of the one shot, these applications involve pulse generation, timing and sequencing. To generate a pulse of specific width, all that the designer needs to do is to add a one shot with the appropriate size external resistor and capacitor.

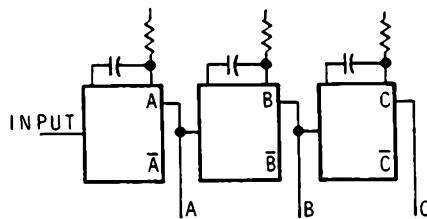


Figure 7-58

One shot pulse sequence generator.

Figure 7-58 shows how one shots can be used for generating a sequence of timing pulses. Here one shots labeled A, B, and C trigger one another. Assume that the one shots trigger on the trailing edge of the input. The waveforms for this circuit are shown in Figure 7-59. When an input pulse occurs, it triggers one shot A. This one shot generates a pulse width (t_1), that is a function of its external component values. At the termination of its output pulse, it triggers one shot B. One shot B generates another output pulse of a specific duration (t_2). Upon its termination this pulse triggers one shot C which produces output pulse (t_3). Such a chain of one shots provides a simple method of sequencing and timing digital operations.

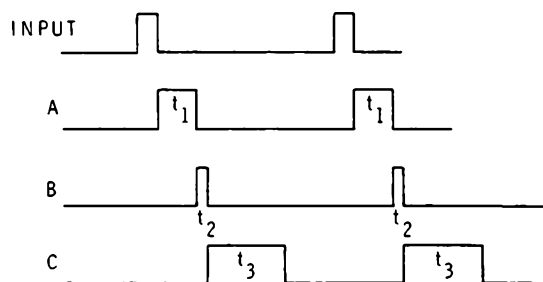


Figure 7-59

Waveforms of the one shot pulse sequencer.

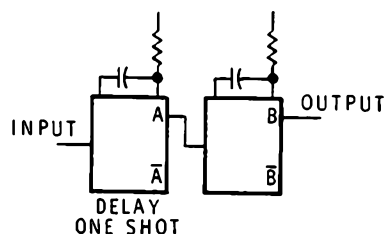


Figure 7-60
Pulse delay using one shots.

Another common application for the one shot is in implementing a delay. In some circuits it is necessary to delay the operation of a particular portion of a circuit. This is essentially a timing operation. A one shot can provide this delay. The input signal to be delayed is applied to the A one shot as shown in Figure 7-60. The A one shot generates the desired delay time. At the end of its delay interval it triggers one shot B which then produces the output pulse that initiates the desired operation. The waveforms in Figure 7-61 illustrate this delay function. The pulse width of one shot B can be adjusted to equal that of the input pulse if desired.

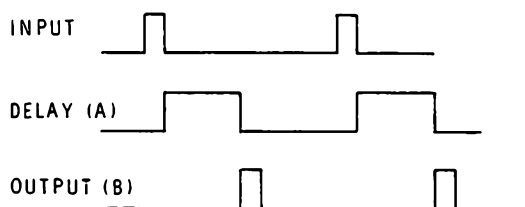


Figure 7-61
Using a one shot
to delay the occurrence of a pulse.

While the one shot appears to be a very flexible and versatile component, it has a poor reputation among digital designers. Before the availability of the high quality integrated circuit one shots, the monostable functions were implemented with discrete component circuits like the one discussed earlier. Such circuits were generally unstable and unreliable. In order to provide a very stable fixed output pulse width, high quality timing resistors and capacitors had to be used. In addition, one shots of this type were very susceptible to false triggering by noise on the power supply line, at the normal trigger input or on the circuit ground. Any stray noise or “glitch” can effectively trigger the one shot and cause timing operations to occur at times when it is not wanted. Because of these problems most digital designers attempt to design without one shots. In most cases, timing functions can be implemented with other types of logic circuits such as counters and shift registers combined with logic gates. In synchronous logic circuits under the control of a master timing clock signal, sequencing pulses with the proper time intervals and durations can be readily generated without the use of one shots. Normally this method is preferred.

The modern integrated circuit one shot has overcome most of the problems associated with the early unreliable circuits. However, the timing pulse stability is still largely a function of the quality of the external resistor and capacitor used to set the output pulse duration. The noise problems have essentially been taken care of by providing high threshold noise immunity at the input. By the use of proper grounding and power supply decoupling networks, false triggering can be kept to a minimum. A good general rule of thumb is to design sequential logic circuits using counters, registers, and gates and developing the timing pulses based on synchronous clock signals. However, you will find some applications where one shots are necessary and desirable.

Self Test Review

45. Most clock oscillators are _____.
46. What determines the frequency of oscillation of most clock circuits?
 - a. Crystal
 - b. Power supply voltage
 - c. RC time constant
47. Two phase clocks are used mainly with which type of logic circuits?
 - a. CMOS
 - b. ECL
 - c. TTL
 - d. MOS
48. A crystal controlled clock is used when the clock frequency must be _____ and _____.
49. A discrete component one shot has a timing resistor of 33 k and a capacitor of .01 μ f. What is the duration of the pulse it generates?
50. The upper duty cycle limit on most one shots is _____ percent.

Answers

- 45. astable multivibrators
- 46. c. RC time constant
- 47. d. MOS
- 48. accurate, stable
- 49. $t = 0.69 (33000) (.01 \times 10^{-6})$
 $t = .2277 \times 10^{-3}$
 $t = .2277 \text{ millisecond or } 227.7 \text{ microsecond}$
- 50. 90 percent

EXPERIMENT 12

Binary Counters

OBJECTIVES:

To demonstrate the operation and characteristics of a binary counter.

Materials Required

Heathkit Digital Design Experimenter
DC Voltmeter
1—74LS04 TTL IC (443-755)
2—74LS76 TTL IC (443-829)
1—74LS193 TTL IC (443-815)

Procedure

1. Construct the four bit binary counter circuit shown in Figure E7-62. The pin connections for the 74LS76 ICs are given in Figure E7-63. Take care in wiring the circuit to avoid errors. Be sure to connect pin 5 to +5 volts and pin 13 to ground (GND) on each 74LS76 IC, and pin 14 to +5 volts and pin 7 to ground on the 74LS04 IC. You will monitor the counter outputs on the LED indicators. You will step the counter with the A logic switch.

Study the counter circuit in Figure E7-62.

What type of counter is this? _____

2. Set the switch SW1 to the high or up position. Apply power to the circuit. Note the states of the LED indicators. Record the binary number stored in the counter.

Note: A (L4) = LSB, D (L1) = MSB

DCBA = _____

3. Depress the B logic switch. Note the states of the LEDs. Record the value of the binary word stored in the counter.

DCBA = _____

Does the change that takes place in the outputs occur on the leading or trailing edge of the \bar{B} signal?

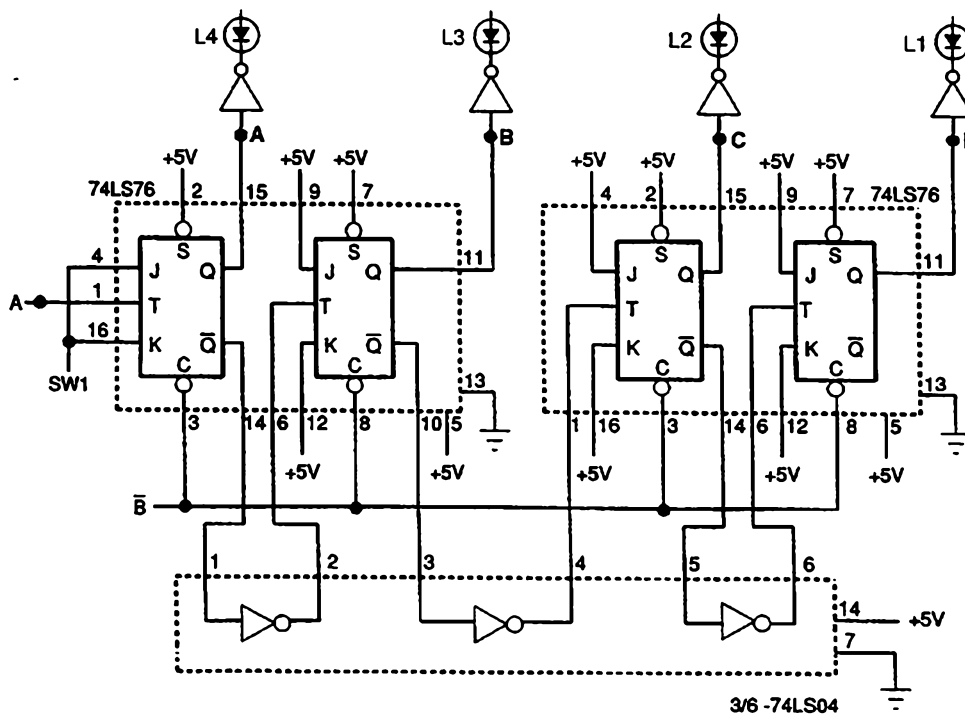


Figure 7-62

Experiment circuit for evaluating the operation of a binary counter.

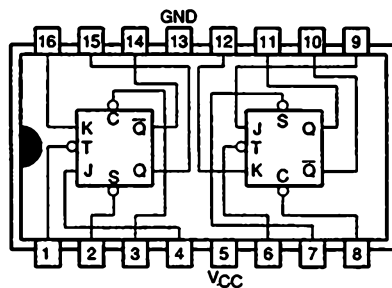


Figure 7-63

Pin connection for the dual JK flip-flop 74LS76.

4. Record the initial counter state obtained in Step 3 in the first (0) position of Table I. Using the A logic switch, step the counter. Each time you actuate the A logic switch, observe the LED outputs and record the counter contents in Table I. Does the counter state change when you depress or release the A logic switch? _____. What does this tell you? _____
5. Observe your data in Table I. What kind of counter did you construct? _____. Does this data confirm your answer given in Step 1? _____. When the counter content is DCBA = 1111, what happens when you _____

depress the A logic switch? _____. The counter output becomes what?

DCBA = _____

6. Remove the counter input from the A output and connect it to the CLK output. Set the clock frequency to 1 Hz. Depress the B logic switch and hold it. Observe the LED indicators. Then, release the B logic switch and let the counter count. As it counts slowly, verify its outputs against your data in Table I. Let the counter run until you fully understand the count sequence.
7. As the counter is counting, set data switch SW1 to the low or down position and observe the result. Repeat several times to be sure you understand what happens. What effect does SW1 have? _____. Depress the B logic switch while the counter is counting. What happens? _____. Return the counter input to the A output of the logic switch.

TABLE I

	D	C	B	A
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Discussion of Steps 1 through 7

In Step 1 you constructed a four-bit binary counter using JK flip-flops. This is a binary up counter of the asynchronous or ripple type since the normal output of one flip-flop is connected to the toggle (T) input of the next flip-flop.

When you apply power to the circuit, the flip-flops can come up in any random state. In Step 3 you used the B logic switch to reset the counter. The \bar{B} output normally rests high so that it has no effect on the asynchronous clear inputs (C) of the flip-flops. When the switch is depressed, \bar{B} goes low thereby putting all flip-flops into the binary 0 state. The counter resets on the leading edge of the \bar{B} signal.

Next, you stepped the counter with the A logic switch, noted the output states on the LED indicators and recorded them in Table I. The counter is stepped or incremented on the trailing edge of the A input signal. The count input from A is normally low. When you depress the A logic switch, A goes high and a leading edge is generated. When you release the A logic switch, A goes low and a trailing edge is generated. Thus, the counter is incremented. By studying the data in Table I you can see that the counter generates a pure 8421 binary code.

An important observation you made in Step 5 was the recycling of the counter from state 1111 to 0000 when the 16th input pulse was applied.

Next, in Step 6 you let the 1 Hz clock signal step the counter automatically. The states change slowly enough for you to see each one, and thus become familiar with this very common count sequence and the recycling step.

You should have found in Step 7 that you could stop the counter from counting in two ways. First, by setting data switch SW1 to the binary 0 position, the counter stops. What you did was to make the J and K inputs of the A flip-flop binary 0 and thereby inhibit its operation. Because the counter is the ripple type, if A doesn't toggle, neither will any of the other flip-flops. The counter simply retains the last

11. Convert the binary numbers you recorded in Table II into their decimal equivalents and record them in the far right hand column of Table II.
12. Observe your data in Table II. What kind of counter is this? _____. Does this confirm your answer in Step 8? _____.
13. Connect the counter input to the CLK output (1 Hz) as you did in Step 6. Let the circuit count. As you do, observe the output states and compare them to your data in Table II. Let the counter run for a while until you see the count pattern or sequence.
14. While the counter is stepping, set the SW1 logic switch to binary 0. Note the result. Next, while the counter is stepping, depress the B logic switch. What happens in each case? _____

Discussion of Steps 8 through 14

In Step 8 you constructed a binary down counter. The complement output of each flip-flop is connected to the T input of the next in sequence. As input pulses are applied, the counter is decremented. Each input pulse decreases the number in the counter by one. When the count is reduced to 0000, it will recycle to its maximum count (1111) when the next clock pulse is applied.

You should have found in Step 14 that this down counter circuit responds to the B logic switch (reset) and SW1 data switch exactly like the binary up counter.

Procedure (continued)

15. Wire the counter circuit shown in Figure E7-65. Use a 74LS193 IC (443-815). Be sure to connect +5 volts to pin 16 and ground to pin 8. The pin connections for the 74LS193 IC are shown in Figure E7-66. You will step the counter with the A logic switch and reset it with the B logic switch. The data switches SW1 through SW4 serve as a parallel data source for presetting the counter.
16. Apply power to the circuit. If the state of the counter is other than 0000, reset it with the B logic switch. Step the counter 16 times with the A logic switch. Note the output states on the LED indicators after each step. Compare the states to those you recorded in Table I.

What type of count sequence does this IC counter generate? _____

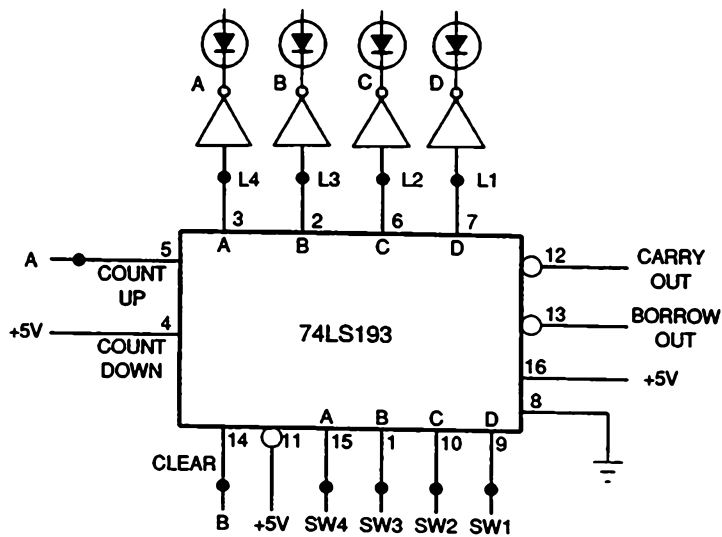


Figure E7-65
Counter circuit using a 74LS193 IC.

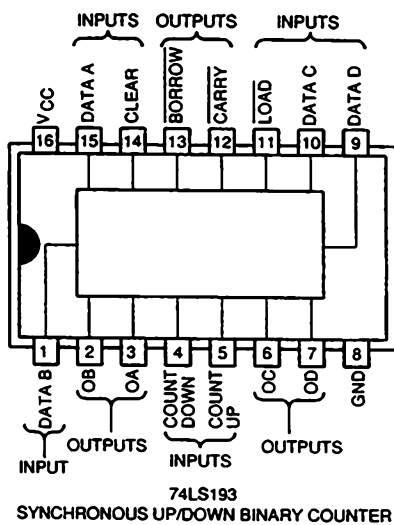


Figure E7-66
Pin connections for 74LS193 IC
binary up/down counter.

17. Reverse the wires at pins 4 and 5 of the IC. Reset the counter with the B logic switch. Apply 16 pulses with the A logic switch and observe the counter output states. Compare them to the data you recorded in Table II. What kind of count sequence is generated? _____.

18. Remove the wires connected to pins 4 and 5 of the IC. Connect pin 4 to +5V and pin 5 to CLK. Be sure the clock frequency is set to 1 Hz. Connect a DC voltmeter between GND and pin 12 of the IC. Set the meter for a reading in the 0 to +5 volt range.
19. Note the voltmeter reading as the counter is stepped by the clock. Record below. At some point in the count cycle, the voltage at pin 12 will change momentarily. When it does, note the new output voltage and the binary state of the counter during the change. Record below.

Voltage at pin 12 before the change (during counting) _____ volts.
 Voltage at pin 12 after the change (momentary) _____ volts.
 Binary Code DCBA = _____ (during momentary change)

20. Reverse the wires at pins 4 and 5 of the IC. Connect the voltmeter to pin 13. Repeat step 19. Record the data below.

Voltage at pin 13 before change (during counting) _____ volts.
 Voltage at pin 13 after change (momentary) _____ volts.
 Binary Code DCBA = _____ (during momentary change)

21. Remove the wires at pins 4 and 5 of the IC. Connect pin 4 to +5V and pin 5 to the A output of logic switch A. Remove the wire between B and pin 14 on the IC. Connect pin 14 to ground. Connect \bar{B} to pin 11. The DC voltmeter can also be removed at this time.
22. Set all of the data switches (SW1–SW4) to binary 1. Depress the B logic switch. Note the output state of the counter.

DCBA = _____

Next, set all data switches to binary 0. Depress the B logic switch and observe the LED indicators.

DCBA = _____

Set the data switches to the words as follows. After each word is set into the switches, depress the B logic switch and note the counter state of the LED indicators. Compare this state to the data switch settings.

SW1	SW2	SW3	SW4
0	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0

What conclusion can you draw from this step? What function is taking place?
_____.

23. Leave the data switches set to 0110. Depress the B logic switch. Note the counter state. Then start stepping the counter with the A logic switch. What happens? _____.

Discussion of Steps 15 through 23

The 74LS193 is a TTL, MSI, up/down counter. It operates synchronously and can be preset (parallel loaded) from an external 4 bit data source. In Step 15 you wired this IC as an up counter. The counter is cleared when B switches from low to high. The counter is incremented when the A logic switch is actuated. The state change occurs on the binary 0 to binary 1 (leading edge) transition of the A signal. As you determined by observing the outputs, the 74LS193 counts in pure binary code. The outputs should be identical to those you recorded in Table I.

In Step 17 you applied the logic switch A count pulse to the down count input. With this connection, the counter is decremented each time you press the logic switch. Your count sequence should have been identical to the sequence you recorded in Table II.

In Steps 18, 19, and 20, you determined the operation of the carry and borrow outputs. The counter was stepped by the 1 Hz clock and you monitored the outputs with a DC voltmeter. During the up count sequence in Step 19, the carry output should have been high (about +3.5 volts). When the 1111 state occurs, the carry output should go low (about +0.1 volt) momentarily. The carry output indicates that the maximum counter value has been reached.

In the down count sequence in Step 20, you monitored the borrow output at pin 13 with the voltmeter. This output should also be high during the count. But when the 0000 state is reached, the borrow output goes low momentarily. The borrow output detects the minimum counter value. To cascade 74LS193 counters, the carry and borrow outputs of one counter are connected to the up and down count inputs, respectively, of the next counter in sequence.

In Steps 21, 22, and 23 you demonstrated how the counter could be preset. The data switches served as your 4-bit parallel source, and you wired the B logic switch to the load input control (pin 11). You should have found that the counter state became the same as the data switch state when the B logic switch was depressed. You parallel loaded the data switch word into the counter. The first important point to remember is that the counter assumed the state of the parallel inputs regardless of its previous contents. In other words, you did not have to reset the counter prior to presetting it. Second, the loading occurs when the load input at pin 11 goes low.

Finally, in Step 23 you stepped the counter after presetting it to 0110. Each actuation of the A logic switch should have incremented the counter. This illustrates that the count sequence simply starts at the preset point and continues in the normal binary sequence. The same applies for down counting.

Remember these operating details of the 74LS193 counter as you will use it later in demonstrating counter applications.

EXPERIMENT 13

The BCD Counter

OBJECTIVE: To demonstrate the operation of an integrated circuit BCD counter.

Materials Required

Heathkit Digital Design Experimenter ET-3200
1 — 74LS90A (443-813) TTL BCD counter

Procedure:

1. Wire the circuit shown in Figure 7-67. You will use a 74LS90A TTL MSI decade counter. You will step the counter from logic switch A. The counter will be reset by using data switch SW1. You will demonstrate the preset to nine operation using data switch SW2. The counter states will be shown on the LED indicators. Be sure to connect +5 volts to pin 5 and ground to pin 10 on the IC. The pin connections for this device are shown in Figure 7-68.

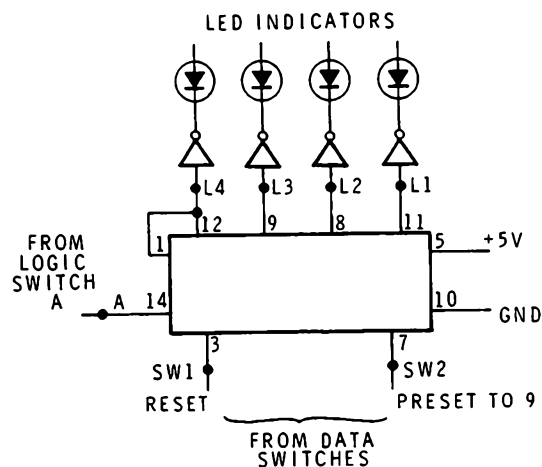


Figure 7-67
Circuit for BCD counter
demonstration.

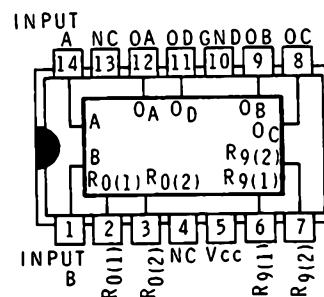


Figure 7-68
Pin connections 74LS90A counter.

TABLE III

D	C	B	A	Decimal

- Be sure the data switches SW1 and SW2 are in the binary 0 position. Next, apply power to the circuit. Observe the states on the LED indicators. Momentarily move SW1 to the binary 1 position and then back to binary 0. Note the effect on the counter state. Record the number in the binary counter before and after you actuate data switch SW1.

DCBA = _____ (before)

DCBA = _____ (after)

- In the first position of Table III, record the counter state that you observed after actuating SW1 in Step 2 above. Then step the counter with the A logic switch. After each actuation of logic switch A, note the LED indicator states and record them in Table III. Apply a total of ten input pulses with the A logic switch and complete Table III. Note particularly the counter state change when the 10th input pulse is applied.
- Convert the binary numbers you recorded in Table III into their decimal equivalent and write them into the spaces provided in Table III. Then observing the data in Table III, verify the operation of the counter. What type of counting function does this IC perform?

- Momentarily move SW1 to binary 1 position and return it immediately to binary 0. Then set SW2 to the binary 1 position momentarily and then return it to binary 0. Note the effect on the output state and record the counter contents below.

DCBA = _____ (SW1)

DCBA = _____ (SW2)

Alternately set SW1 then SW2 to binary 0 several times to be sure that you see what effect that these two inputs have.

- Remove the wire connecting pin 14 of the IC to the output of the A logic switch. Connect pin 14 to the CLK output. Set the clock frequency to 1 Hz. Let the clock step the counter. Observe the counter steps and follow the sequence by referring to the states you obtained and recorded in Table III. Let the counter cycle for awhile to be sure that you see and understand the count sequence that it produces.
- Remove the connections from the counter to LED indicators L2, L3, and L4. Only indicator L1 should be connected to the IC counter. When you're removing these connections, be sure to retain the connection between pins 1 and 12 on the IC. The L1

indicator that you have left connected is monitoring the most significant bit of the counter. Remove the connection between pin 14 on the IC and the CLK output. Connect pin 14 to the A output terminal of logic switch A.

8. Momentarily set SW1 to the binary 1 position. Then apply logic count pulses to the counter by actuating the A logic switch. Count the number of pulses that you apply to the circuit. While you are doing this, monitor the output state of L1. Each time that L1 turns on and then off, indicate its occurrence by marking a 1 in the margin of the page. At that time also note the number of input pulses that have been applied to the counter up to that point. Each time L1 turns on and then off, start the input count over again. How many input pulses occur for each single output pulse? _____.

Discussion

In this experiment you demonstrated the operation of the 7490A TTL MSI BCD counter. As you should have discovered from evaluating the count sequence you recorded in Table III, this circuit counts in the standard 8421 BCD code. When the circuit reaches its maximum count of 9 (1001) the next input pulse causes the counter to recycle to 0000. The states 1010 through 1111 are invalid in a BCD counter.

You used data switch SW1 to reset the counter to 0. When power is first applied to the counter it can come up into any state. By momentarily putting SW1 in the binary 1 position, the counter should reset to 0.

You demonstrated how data switch SW2 could be used to preset the counter to 9. When SW2 is momentarily moved to the binary 1 position, the counter is preset to 9 (1001). The preset to 9 operation is not a widely used counter function. However, it is used in some applications where certain arithmetic operations with BCD numbers are carried out with the counter.

In Steps 7 and 8 you demonstrated that the frequency counter divides by 10. Such a counter is generally referred to as a decade counter. The divide by 10 action is demonstrated by the fact that you should have recorded a change in the L1 output indicator for every 10 input pulses. With the counter starting at binary 0, the L1 indicator which monitors the D flip-flop output remains reset for the first 7 input pulses. On the eighth input pulse, the L1 indicator lights indicating that the D flip-flop has been set. Upon application of the 10th input pulse, the L1 indicator switches off. As you can see from the truth table you developed in Table III, the D flip-flop is set for two counts. The D flip-flop sets and then resets every 10 input pulses.

EXPERIMENT 14

Counter Applications

OBJECTIVE: *To demonstrate several practical applications for binary and BCD counters.*

Materials Required

Heathkit Digital Design Experimenter

1 - 74LS00 TTL IC (443-728)

1 - 74LS04 TTL IC (443-755)

2 - 74LS76 TTL IC (443-829)

1 - 74LS90 TTL IC (443-813)

1 - 74LS193 TTL IC (443-815)

Procedure

1. Refer to the experimental circuit in Figure E7-69. This is a scaler circuit using counters. Study the circuit and determine the ratio by which it divides the input frequency. In the spaces provided below, record the input frequency and the frequencies at points X, Y, and Z in the circuit. Refer back to the appropriate sections in the unit or previous experiments, if necessary, to determine how the circuit operates.

Frequency at: INPUT _____ Hz
X _____ Hz
Y _____ Hz
Z _____ Hz

2. Construct the circuit shown in Figure E7-69 (on the ET-1000 Trainer, the input is taken from the square wave generator set to 60 Hz.) As before, take your time to be sure that the circuit is wired correctly. As the experiments become more sophisticated, the number of integrated circuits to be interconnected increases. This also increases the chances of your making a wiring mistake. If the circuit does not perform properly, the first thing to check is your circuit wiring. Be sure that you have connected +5 volts and ground to each of the integrated circuits in the experimental circuit.
3. To verify the operation of the circuit, apply power and observe the outputs Y and Z on LED indicators L3 and L4, respectively. The frequency of the output pulses at Z will be slow enough for you to count them. Count the number of

pulses at Z that occur in one-half minute using the second hand on your watch as a timing indicator. Record the number of pulses occurring in a half minute.

What is the overall frequency division ratio of this circuit? _____

What is the basic function of this circuit? _____

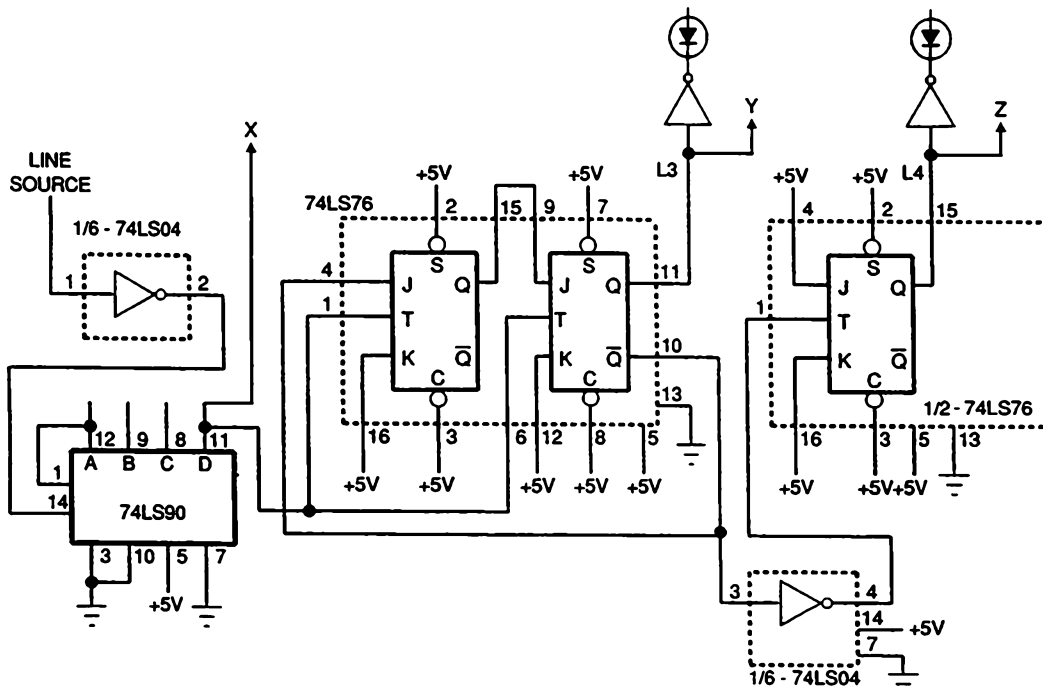


Figure E7-69

Scaler circuit for Steps 1, 2, and 3.

4. Refer to the circuit shown in Figure E7-70A. Study this circuit carefully, noting the function of each logic element in the circuit. Analyze the operation of the circuit. To do this, assume that the 74LS193 counter is initially reset. Also, assume that the latch made up of gates 1 and 2 is also initially reset so that the output at pin 6 of gate 2 is binary 0. Further assume that the operation of the circuit is started by actuating the B logic switch a single time. Finally, assume that the data switches are set so that SW4 = 1, SW3 = 0, SW2 = 1, and SW1 = 0. Sketch the output of gate 4.

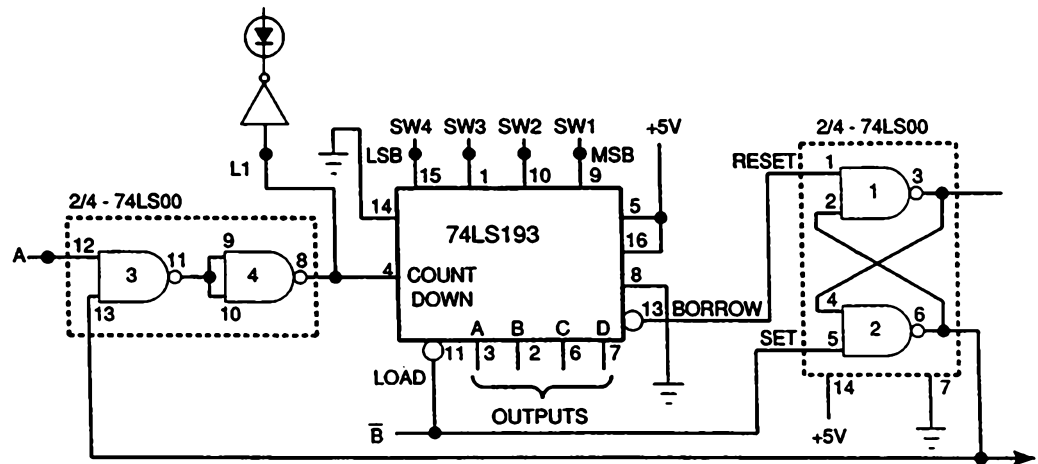


Figure E7-70A
Experimental circuit for Steps 4 and 5.

5. Construct the circuit shown in Figure E7-70A. The circuit will be driven by logic switch A. You will observe the output of the circuit on LED indicator L1. Set the data switches as indicated in Step 4.
6. Apply power to the circuit. L1 should be off at this time. To start the operation of the circuit, depress logic switch B once. Then, while observing the L1 output, depress logic switch A several times. Count the number of pulses that occur on L1. Keep depressing logic switch A until L1 stops pulsing. How many output pulses occur before L1 remains off? _____

Set the data switches so that SW4 = 0, SW3 = 0, SW2 = 1, and SW1 = 1. Record below, the binary and decimal numbers represented by this word. Momentarily depress logic switch B. Depress logic switch A a number of times, again, count how many pulses occur at L1 before the pulses stop.

Binary number = _____
 Decimal number = _____
 Pulses at L1 = _____

7. Using the information that you collected in Step 6, compare the number of output pulses occurring on L1 with the decimal value of the binary number loaded into the 74LS193 counter. How are they related? What is the basic function of this circuit? _____

Discussion

The circuit shown in Figure E7-69 is a divide-by-60 scaler. It accepts the 60 Hz waveform from the line source output on the Trainer and divides it by 60. The Z output is $1/60$ of the input frequency, or 1 Hz.

The inverter (74LS04), connected between the line source and the input to the 74LS90, serves only to buffer the ET-1000 Trainer's square wave generator output. If you are using any other trainer, the inverter is not required.

Close analysis of this circuit will show that the output at point X is $1/10$ of the input frequency, or 6 Hz. The 74LS90 decade counter is used as a divide-by-10 circuit.

The first 74LS76 IC in the circuit is connected as a divide-by-3 (modulo 3) counter. The output at point Y then is $1/3$ of the frequency at point X or $6 \div 3 = 2$ Hz. This signal is applied to one of the JK flip-flops in the other 74LS76 IC. It divides the frequency of point Y by 2 to produce an output frequency of 1 Hz. The signal at Z is a 1 Hz square wave with a 50 percent duty cycle.

Because the line source in the Trainer is derived from the 60 Hz power line, the input frequency is very accurate. The frequency of the power line voltage has an error typically less than 0.1 percent. For that reason, the output frequency at Z is a very accurate 1 second source. With a 50 percent duty cycle at the output, LED indicator L4 remains on for one-half second and then off another half-second.

When you counted the number of output pulses on L4 occurring in one-half minute, you should have recorded a value of 30. One-half minute, of course, is equal to 30 seconds and a 1 Hz signal will cause 30 pulses to occur during that period of time.

The circuit in Figure E7-70A is designed to generate a fixed number of output pulses upon command, and then, automatically stop. The number of output pulses to be generated by the circuit is determined by the binary number input set into the data switches. For example, you set in the binary number 0101, which is the binary equivalent of the decimal number 5. When the B logic switch is actuated and the A logic switch is pulsed, the circuit will generate 5 output pulses and then stop. These output pulses occur at the rate logic switch A is pulsed. You observed them on LED indicator L1. The waveforms produced by this circuit are illustrated in Figure E7-70B.

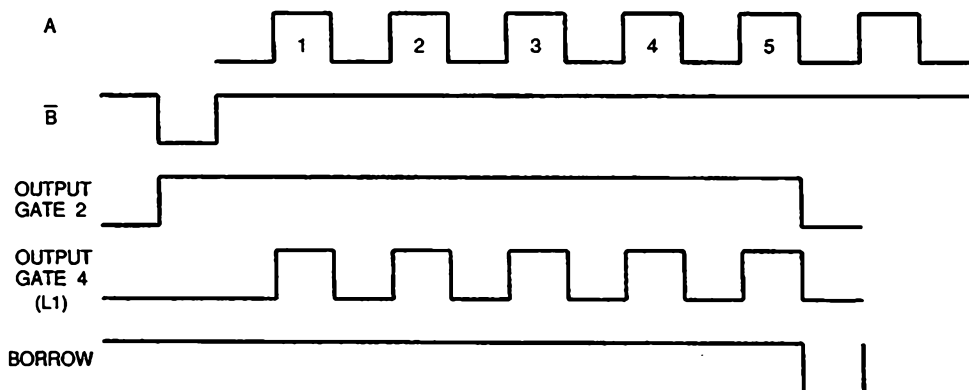


Figure E7-70B
Waveforms for the circuit in Figure E7-70A.

Here's how the circuit operates.

Assume that the 74LS193 binary counter is initially reset. Note that it is wired as a down counter—the count pulses are applied to the down count input at pin 4. Another clue that this device is being used as a down counter is the use of the borrow output. The borrow output is effective only in the down counting mode. Assume also that the latch circuit, made up of gates 1 and 2, is also reset. This means that the output of gate 2 is low. This low output inhibits the AND gate, made up of gates 3 and 4, in the 74LS00 IC. The pulses from logic switch A step the counter only when gate 3 is enabled.

The signal from the B logic switch is used to initiate the circuit operation. When the B logic switch is depressed, the \bar{B} output goes low. This signal causes two things to happen. First, it forces the load input on the 74LS193 low, thereby presetting the counter to the binary number set on the data switches, in this case, 0101. At the same time, this signal sets the latch made up of gates 1 and 2. With the latch set, the output of gate 2 is high. Gate 3, therefore, is enabled and pulses pass through gate 3 and gate 4 (which is connected as an inverter) and cause the counter to step. The counter counts down, starting at its preset number 5. The counter continues to step until the 0 condition is reached. At this time, the borrow output line goes low causing the latch to reset. As the latch resets, gate 3 is again inhibited and the pulses no longer reach the counter.

During the time that the counter is decrementing, LED indicator L1 monitors the pulses occurring at the output of inverter 4. Because it takes five pulses to decrement the counter to 0, this LED indicator will switch off and on five times. This indicates that five output pulses occurred, corresponding to the binary number preset into the counter.

You again demonstrated the operation of the circuit by programming the counter with the number 1100 or decimal 12. When the B logic switch is actuated, the circuit generates 12 output pulses before it stops.

The number of output pulses generated by this circuit is limited by the count capability of the 74LS193 counter. Its maximum count is 1111 or 15. In order to extend this to larger values, additional 74LS193 counters can be cascaded to accommodate the desired number of pulses.

EXPERIMENT 15

Shift Registers

OBJECTIVES: *To demonstrate the operation and characteristics of bipolar integrated circuit shift registers.*

Materials Required

Heathkit Digital Design Experimenter ET-3200

1 – 74LS04 IC (443-755)

2 – 74LS76 IC (443-829)

1 – 74LS95 IC (443-814)

Procedure

1. Construct the four bit shift register circuit shown in Figure 7-71. You will use two type 74LS76 dual JK flip-flops. Shift pulses for the circuit will be derived from logic switch A. Logic switch B is connected to reset the shift register via the asynchronous clear inputs on the flip-flops. The serial input to the shift register will come from data switch SW1. Note that one of the inverters in a 74LS04 IC is used to make the JK inputs of the first flip-flop complementary. The shift register outputs will be monitored on the LED indicators. Don't forget to connect +5 volts and ground to each IC.

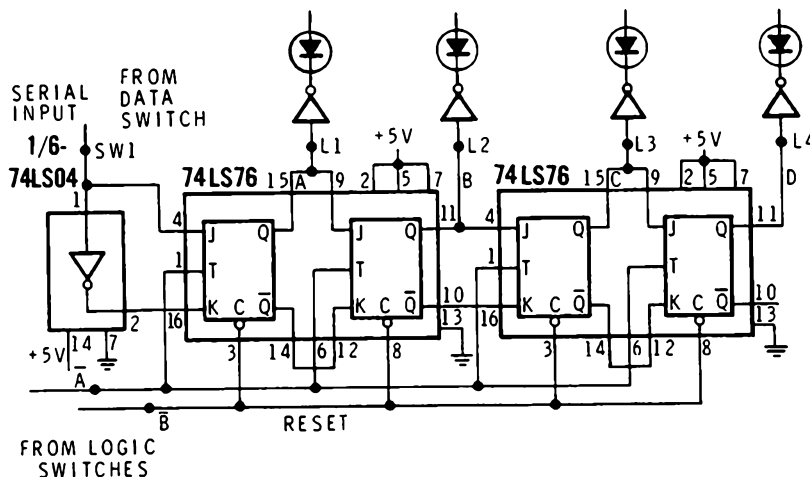


Figure 7-71
Shift register demonstration circuit.

2. Apply power to the circuit. Reset the shift register by actuating the B logic switch. Set data switch SW1 to the binary 1 position. Then using the A logic switch, apply 4 shift pulses. Observe the LED indicators as you do. Record the binary value of the register contents after four shift pulses.

ABCD = _____.

Next, set the SW1 switch to binary 0. Again apply four shift pulses with the A logic switch. Again record the binary contents of the register.

ABCD = _____.

3. Using the SW1 data switch and the A logic switch, load the shift register by using the step by step procedure given below.

SW1 = 1, depress switch A

SW1 = 0, depress switch A

SW1 = 1, depress switch A

SW1 = 0, depress switch A

After you have loaded the shift register, observe the LED indicators and in the space below write the decimal equivalent of the binary number in the shift register.

Decimal number = _____.

Discussion of Steps 1 through 3

In these steps you constructed a four bit shift register with JK flip-flops. You loaded data into the shift register serially using data switch SW1 as your data source. A single bit of information was entered into the shift register with each actuation of the A logic switch. As the bits were entered into the A flip-flop, they were shifted to the right one bit at a time for each shift pulse.

The 74LS04 inverter is used to convert the data from SW1 into two complementary signals which are applied to the JK input of the input (A) flip-flop. The JK inputs must always be complementary in a shift register using a JK flip-flop.

In Step 2 you applied a binary 1 to the shift register input. Then actuating the A logic switch four times, you generated four shift pulses which loaded the shift register with binary 1s. As you generated the shift pulses, you should have noted the entry of the first binary 1 bit into the A flip-flop and with the second shift pulse, the shifting of the data to the right until the register was full (1111). Next, you set the input to binary 0 and then loaded the register with binary 0s with four shift pulses (0000). As the binary 0s were loaded, the binary 1 bits in the register were shifted out.

In Step 3 you loaded the binary number 0101 into the register a bit at a time by setting SW1 to the desired state and then depressing a logic switch to generate a shift pulse. While the bit pattern in the shift register is easy to identify by simply observing the LED indicators, you cannot convert that bit pattern into its decimal number equivalent unless you know which bit is the LSB. Since this information was not given, your answer could have been either $0101 = 5$ or $1010 = 10$. In a binary counter the least significant bit is readily identified since it is the flip-flop to which the input or count pulses are applied. With this information you can always determine the value of the number in the counter. With the shift register, however, the input flip-flop may not necessarily be the least significant bit. We could also assign the right most or serial output flip-flop as the least significant bit. The choice of weight for the input and output flip-flops is up to the designer and will depend upon his application. For a large percentage of applications, the input flip-flop contains the most significant bit while the output flip-flop contains the least significant bit. In other words, data is entered into the shift register beginning with the least significant bit. As data is shifted out of the shift register the least significant bit is shifted out first. Unless otherwise stated, we will use that convention here. As you can see from the demonstration described here, the shift register does perform a serial to parallel conversion. Data is entered serially from the SW1 data switch and then displayed in parallel on the LED indicators. Data can also be entered serially with SW1 and read-out serially by simply observing the state of L4.

Procedure (continued)

- Construct the shift register circuit shown in Figure 7-72. At this time you can disassemble the shift register circuits used in the previous steps. The shift register shown in Figure 7-72 is a 74LS95 IC. This is an MSI shift register already completely wired internally and ready to use. The flexibility of the input/output leads permits this device to be used for a wide variety of functions. It eliminates the need to interconnect individual flip-flops to perform shift register operations. As before you will use the LED indicators to observe the shift register contents. Data switches SW1 through SW4 will be used as a parallel data source for the shift register. Logic switch A will be used to generate the shift pulses in the previous steps. Logic switch B will serve as a mode control for the circuit. Don't forget to connect +5 volts and ground to the 74LS95 IC.

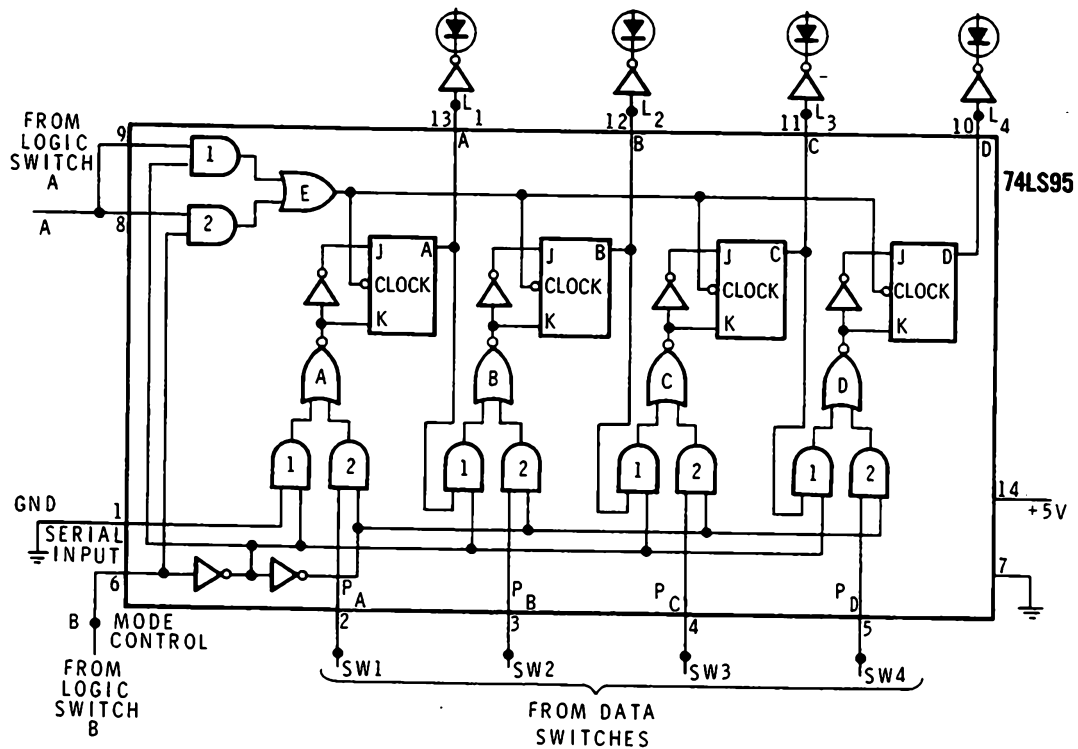


Figure 7-72
74LS95 MSI shift register
circuit for Steps 4 and 5.

5. Apply power to the circuit. As before, the flip-flops in the shift register can come up into any state. Disregard the LED indicator states at this time. Set all of the data switches to binary 0. Then depress the B logic switch and hold it in the down position. Then momentarily depress the A logic switch. Note the LED indicator states and record below.

DCBA = _____.

Next, set all of the data switches to binary 1. Again depress the B logic switch. Holding it in the low position, actuate the A logic switch momentarily. Release both switches. Observe the LED indicator states and record the binary number below.

DCBA = _____.

Depress the A logic switch four times and note the LED indicator states. After you have applied four shift pulses, record the LED indicator states below.

DCBA = _____.

Discussion of Steps 4 and 5

In Step 5 you demonstrated how a shift register can be preset by parallel loading it from some data source. In this case the data source was the data switches SW1 through SW4. First, you set the data switches to binary 0. You then loaded the binary number 0000 into the register. You did this by setting the mode control high by depressing the B logic switch. Then you generated a single clock pulse by depressing the A logic switch. The A logic switch generates a clock pulse that causes the parallel input to be preset into the register.

The 74LS95 shift register has a mode control input line terminated at pin 6. When this mode control input line is binary 0, the shift register is set up for shift right operations. When the mode control input line is binary 1, the shift right function is disabled and the circuitry for parallel loading the shift register from an external source is enabled. What you did when you depressed the B logic switch is to set the mode control line high. Then you actuated the A logic switch to generate the clock pulse that actually loads the register.

Next, you set the data switches to binary 1. Again, using the A and B logic switches you preset the register. The number in the register at this time should be 1111. Finally, you applied four shift pulses with the A logic switch. This caused the binary 1111 to be shifted out serially to the right. After the four shift pulses were applied, the register contents should have been 0000 since the serial input (pin 1) is low. In this step then you demonstrated how the shift register can be parallel loaded and how this parallel data can then be shifted out serially. This demonstrated the parallel to serial data conversion process performed by a shift register.

Procedure (continued)

6. Modify your 74LS95 shift register circuit to conform to the configuration shown in Figure 7-73. Here you are connecting the parallel data inputs back around to the outputs in order to permit the shift register to perform shift left operations. As before you will use the A logic switch to generate shift pulses. Data switch SW1 will be used for a serial data input for shift right operations. SW4 will be used as the data source for shift left operations. Switch SW2 will be used to control the mode of the circuit. The mode control will select either shift right or shift left operations.

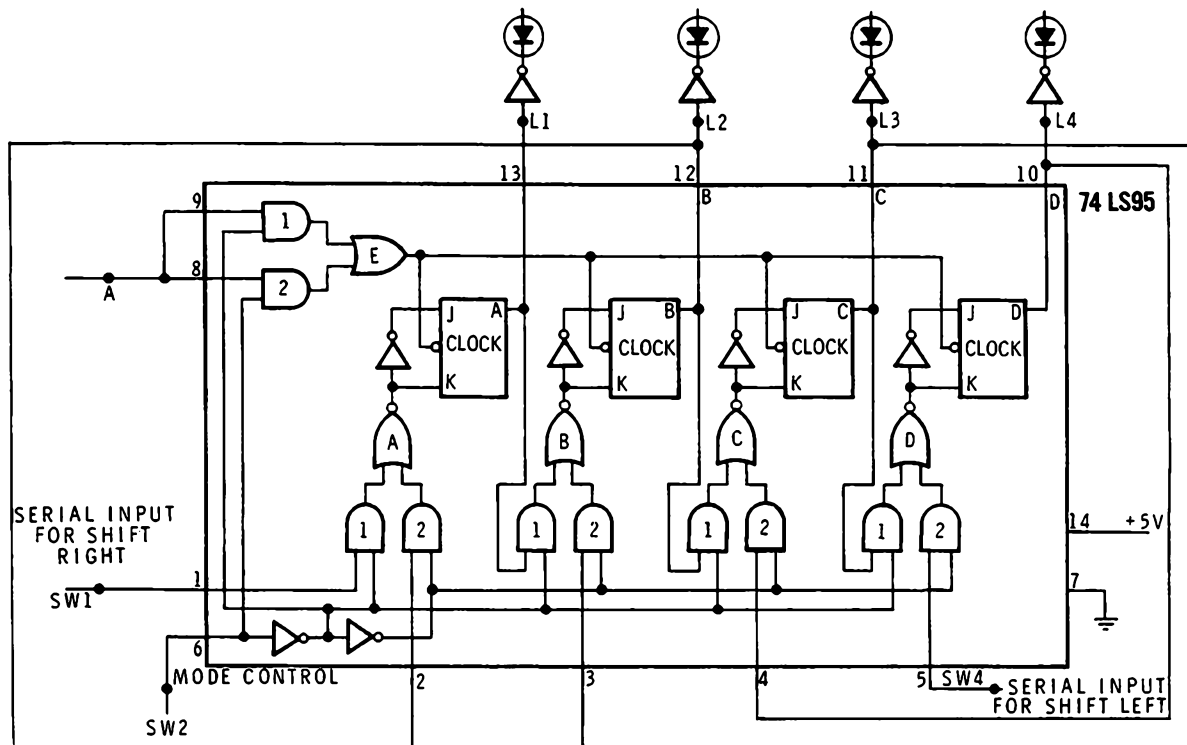


Figure 7-73
Shift right/shift left
circuit for Steps 6, 7, and 8.

Studying the circuit in Figure 7-73, determine the binary state of the mode control input to perform shift left operations.

_____.

7. Set data switches SW1, SW2, and SW4 to the binary 0 state. Apply power to the circuit and depress the A logic switch four times. Record the state of the LED indicators below.

ABCD = _____.

Next, set switch SW1 to the binary 1 position. Depress the A logic switch four times. Note the direction of shifting as the shift pulses are applied. After four pulses have been applied, record the state of the register below.

ABCD = _____.

8. Set SW1 to binary 0. Again depress the A logic switch four times. Note the direction in which the data shifts. Next, set SW2 to binary 1, and the SW4 switch to binary 1. Apply four shift pulses with the A logic switch. Again note the direction of shifting, and record the contents of the register below.

ABCD = _____.

Set the SW4 switch to binary 0 and apply two shift pulses. Note the direction of the shifting and record the LED indicator states in the space below.

ABCD = _____.

Discussion of Steps 6, 7, and 8

In these steps you demonstrated how the 74LS95 IC shift register can perform both shift right and shift left operations. In Step 7 you shifted data to the right using SW1 as the serial input data source. First, you shifted in binary 0s to clear the register then shifted in binary 1s. You then shifted out the binary 1s as binary 0s were shifted in. During these shifting operations you should have noted that the data moves from left to right. The logic indicators have been wired so that they indicate the direction of shift directly. Data moves from flip-flop A to flip-flop B to flip-flop C to D or from logic indicator L1 to L2 to L3 and finally to L4. All this occurs with the mode control in the binary 0 position.

When the SW2 switch is placed in the binary 1 position, the register will be set up for shift left operations. The 74LS95 IC is internally wired to perform shift right operations automatically. By simply enabling the mode control with a binary 0 the shift right function is performed. However, the shift left operation must be externally wired if it is to occur. The shift left operation is implemented by connecting the appropriate flip-flop outputs back to the parallel input lines. The parallel input lines permit the 74LS95 to be either preset from some parallel source or connected for shift left operations. Note in Figure 7-73 that the D flip-flop output is connected to the C flip-flop input. The C flip-flop output is connected to the B flip-flop input. And finally, the B flip-flop output is connected to the A flip-flop input. The D flip-flop receives its input from data switch SW4. This is the serial input line for the shift register when used for shift left operations. Now as shift pulses are applied, data will move from SW4 to the D flip-flop then to C, B, and then A. You demonstrated the shift left effect by loading all binary 1s into the register. As you did you should have seen the LEDs light from right to left indicating a left shift. You then applied two shift pulses with the SW4 switch set to binary 0. This causes binary 0s to be loaded into the C and D flip-flops. The binary 1s stored in those two flip-flops previously are shifted to the A and B flip-flops. Therefore, the binary number stored in the register after the two pulses are applied will be 1100.

This completes your work for Experiment 15. Do not disassemble your shift register circuit as it will be used in the next experiment.

EXPERIMENT 16

Shift Register Applications

OBJECTIVES: To demonstrate several practical applications of integrated circuit shift registers.

Materials Required

Heathkit Digital Design Experimenter ET-3200

1 — 74LS04 IC (443-755)

1 — 74LS95 IC (443-814)

Procedure

1. For this experiment you will use the 74LS95 shift register circuit you constructed in Experiment 15. As you recall the register was wired for both shift right and shift left operations. To perform the next step you do not need to make any further modifications to the circuit. The circuit is repeated in Figure 7-74.

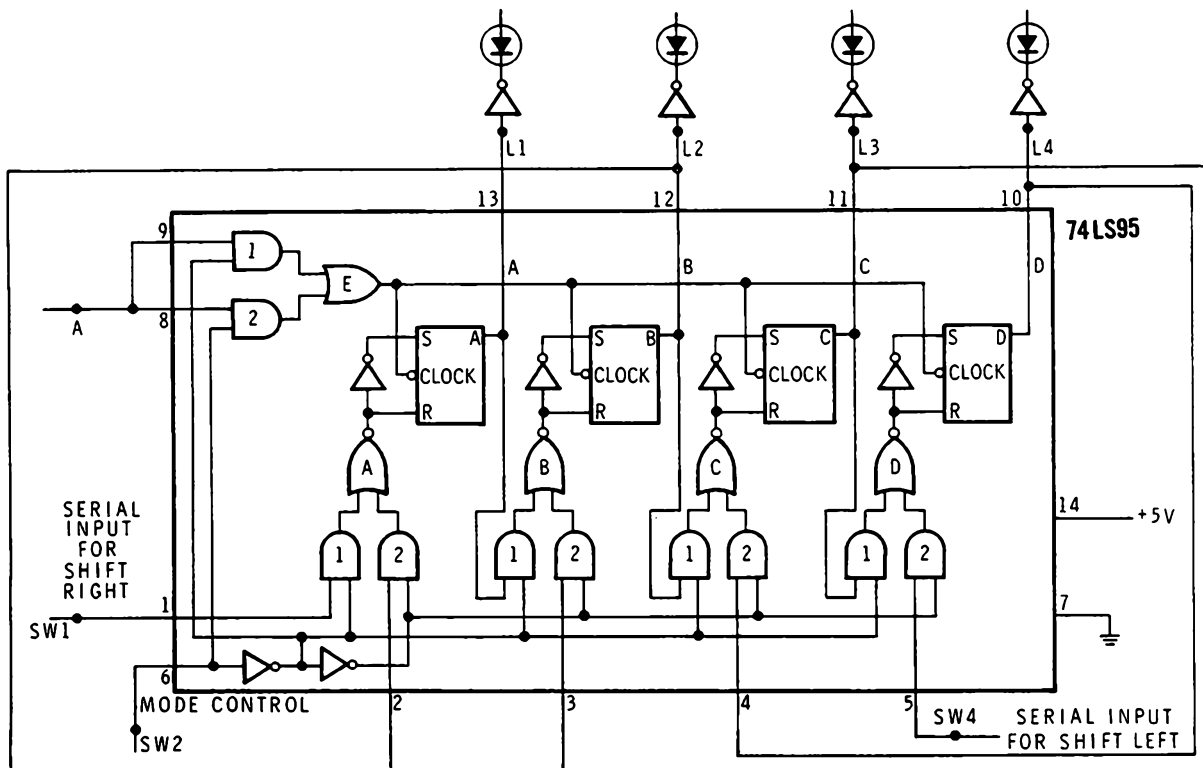


Figure 7-74
Shift right/shift left
circuit for Steps 1 through 5.

2. Set the SW2 switch to binary 1. SW1 should be set to binary 0. Set SW4 to binary 0 and depress the A logic switch 4 times. The register is set up for shift left operations and this step should clear the register to 0.

Next set SW4 to binary 1. Depress the A logic switch two times. Record the binary number stored in the register. Assume that the D flip-flop (indicator L4) is the LSB.

ABCD = _____,
Decimal value = _____.

Now set SW4 to binary 0.

Depress the A logic switch once and again note the contents of the register. Record it and its decimal equivalent below. ABCD = _____, decimal value = _____. Again depress the A logic switch a single time. Record the binary and decimal equivalent of the register contents.

ABCD = _____, decimal value = _____.

3. Study the data you obtained in Step 2 above. Determine the relationship between the numbers obtained when the register was loaded and as it was shifted to the left. What mathematical operation was performed by the shift left operation? _____.
4. Set SW2 and SW4 to binary 0. The SW1 switch should also be at 0 at this time. Clear the register by pressing the A logic switch 4 times. Next, load the register by following the step by step instructions below.

SW1 = 1, Depress the A logic switch
SW1 = 0, Depress the A logic switch
SW1 = 1, Depress the A logic switch
Set SW1 = 0

Record the binary contents of the register and its decimal value below. Again use the D flip-flop (indicator L4) as the LSB.

ABCD = _____, decimal value = _____.

Depress the A logic switch once. Note the binary value of the register contents and record it in its decimal equivalent below.

ABCD = _____, decimal value = _____.

5. Study the data that you obtained by the shift right operations you carried out in Step 4. What mathematical operation is performed when a shift right operation occurs? _____.

Discussion of Steps 1 through 5

In these steps you demonstrated how a shift right/shift left circuit could be used to perform multiplication and division operations on binary numbers. In Step 2 you loaded the decimal number 3 into the shift register. Then you shifted it to the left one step. Evaluating the number in the register you saw that it was 6. Shifting the number one more bit position to the left produced the binary number 12. Your conclusion from this is that with each shift left operation the number stored in the register was multiplied by 2. Here, you shifted the number 3 two positions to the left producing a total multiplication factor of $2^2 = 4$.

Next, you set up the shift register for producing a shift right operation. You initially loaded the binary number 1010 into the register. Of course, this is the binary equivalent of the decimal number 10. You then applied a single shift pulse and caused the number to be shifted to the right. Evaluating the new number you found it to be 0101 or 5. Here shifting a number to the right causes that number to be divided by two for each shift right. The original number in the register is divided by a number equal to 2^N where N is the number of positions shifted to the right.

A shift right/shift left shift register is easy to use for scaling operations involving the multiplication or division of number by some power of 2.

Procedure (continued)

6. Next, you are going to demonstrate how the data in a shift register can be recirculated by feeding the serial output back to the serial input. This permits the data to be shifted out serially for use in some external source but the data will still be retained since it is recirculated.

Modify your 74LS95 shift register circuit so that it appears as shown in Figure 7-75. Note that the output from the D flip-flop is fed back to the serial input of the shift register at pin 1. You will use the data switch SW1 as the mode control and SW4 will serve as an additional serial input. The logic control gating at the input to the A flip-flop permits either of two data sources to be shifted into the register, that from pin 1 and that from pin 2.

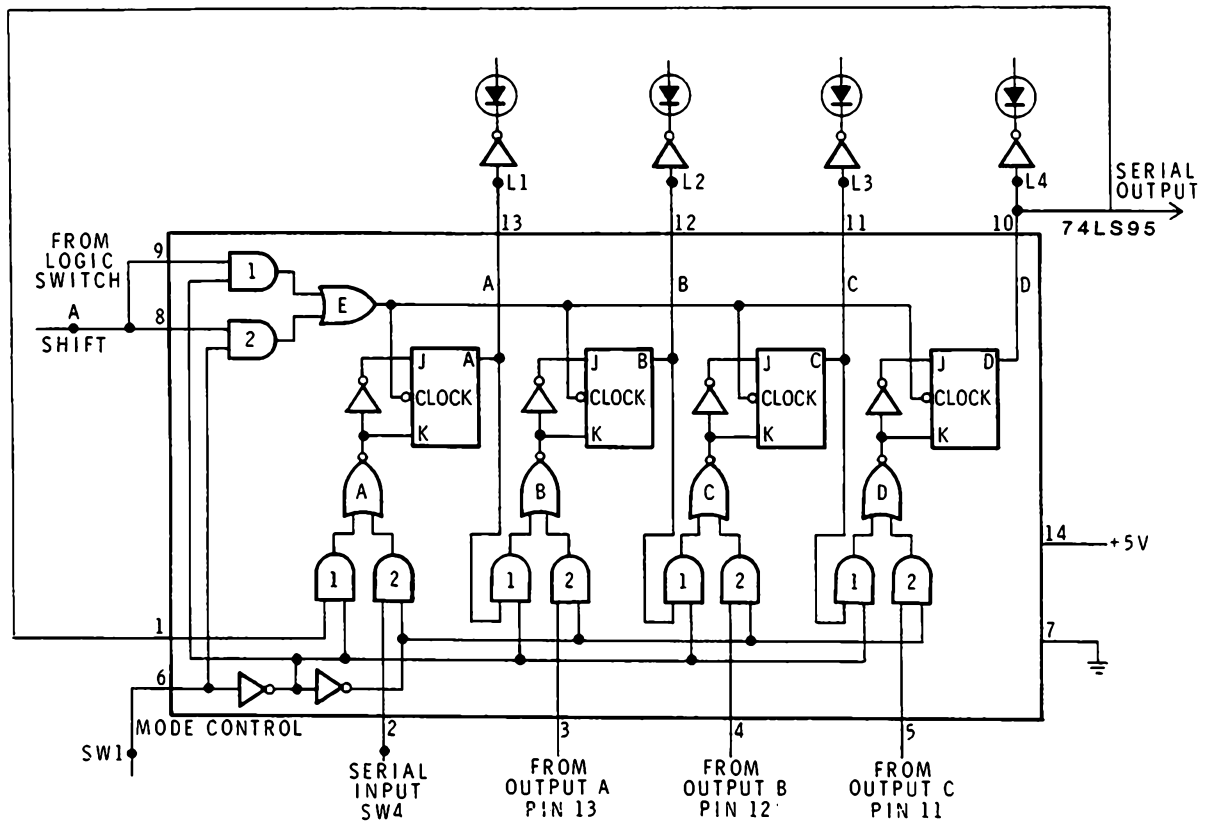


Figure 7-75
Write/recirculate shift
register circuit for Steps 6 through 9.

- Set SW1 to binary 1 and SW4 to binary 0. Depress the A logic switch four times. This should clear the register and the other indicators should be off.

Next, set SW4 to binary 1. Depress the A logic switch two times. In the space provided below, record the binary number stored in the register.

ABCD = _____.

Next, set SW1 to binary 0. Then depress the A logic switch four times. For each shift pulse, note the position of the binary 1 bits on the LED display. Record the state of the register contents after four shift pulses have been applied.

ABCD = _____.

In Table I, record the state of the shift register contents as indicated by the LED indicators. Then after each clock pulse, again record the four bit register state. Complete the table as indicated.

8. Study the information you recorded in Table I. From this information determine the operation of the circuit. Refer back to Figure 7-75 if necessary to see how the circuit operates. Is the data in the shift register lost or retained as a result of shifting the data? _____.
9. Set the SW1 mode control switch to binary 1. Set SW4 to binary 0. Again apply four shift pulses with the A logic switch. Note the contents of the register after the four shift pulses have been applied and record the register state below.

ABCD = _____.

As a result of the operation above, was data lost or retained as a result of the four shift pulses? _____.

TABLE I

A	B	C	D

Discussion of Steps 6 through 9

In these steps you demonstrated how data can be recirculated in the shift register in order that the contents be retained even when the data must be shifted out serially to another source. This is done by connecting the output of the shift register from the D flip-flop back around to the serial input to the A flip-flop at pin 1. With the mode control set to the binary 0 position, the shift register will perform a shift right operation. As the shift right operation is performed, the serial data appears a bit at a time at the normal output of the D flip-flop. But at the same time, this data is shifted back into the shift register. It takes four shift pulses to cause a single four bit binary word to be shifted out. After four shift pulses occur, the data is also shifted back into the register and is ready to be used again.

When the mode control switch is set to binary 1, the serial data input at pin 1 on the 74LS95 IC is disabled. In this case, the input at pin 2 is recognized. This permits an external serial data source to feed data to the shift register. In this case you used SW4 to provide data to the shift register. With the mode control input at binary 1, you can load a serial word appearing at pin 2 into register as shift pulses are applied.

An important point to note is that with the mode control in the binary 1 state the parallel data inputs are enabled. The input to flip-flop A is used as the serial data source. But the other inputs must be connected to flip-flops A, B, and C respectively in order for a shift right operation to be performed with the mode control input high.

In Step 7, you set the mode control to the binary 1 position and the SW4 switch to binary 0. This caused data (0000) to be loaded into the shift register from SW4. Next, you loaded two binary 1s by setting SW4 to binary 1 and depressing the A logic switch twice. This loaded the number 1100 into the register.

Next, you set the mode control input to binary 0. This disables the serial input from SW4 and causes the data to recirculate. As you applied four shift pulses, you should have observed the data shifting right out of the D flip-flop and then back around into the A flip-flop. After four shift pulses, the data is shifted out of the register but it is also recirculated. After four shift pulses, the contents of the register is still 1100. Your data in Table I should appear as shown in Table II.

TABLE II

A	B	C	D
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1

→ DIRECTION OF SHIFT

RECYCLE

Finally, you set the mode control switch to binary 1. This again enables the serial input from SW4. You then loaded binary 0s with four shift pulses. As you loaded these 0s you noticed that the binary number 1100 was shifted out serially and lost as the new number 0000 was shifted in.

A shift register when connected in this way forms what is known as a load/recirculate register. The mode control input line lets you put in data from an external serial source. In the recirculate mode it permits data to be shifted out and used externally but also recirculates it so that it is retained for another operation.

Procedure (continued)

10. Using the recirculate register shown in Figure 7-75, clear the register by loading all 0's. If the register is already at 0 then this operation is not necessary. To clear the register, set SW1 to binary 1 and SW4 to binary 0. Then depress the A logic switch 4 times.
11. Next, set SW4 to binary 1. Depress the A logic switch once. Set the mode control switch SW1 to binary 0. Then begin depressing the A logic switch. Note the result. Continue depressing the A logic switch a number of times until you are fully aware of what the circuit is doing.

Discussion of Steps 10 and 11

In these steps you demonstrated the operation of the shift register as a ring counter. You cleared the register and loaded a binary 1 into the A flip-flop. You then set the mode control so that the shift register would recirculate. Then by depressing the A logic switch you were able to cause the binary 1 bit to move from one flip-flop position to the next and then recirculate. When used in this manner, the shift register is known as a ring counter. This ring counter makes an excellent sequencing circuit for driving digital circuits that require a time sequence of pulses.

Procedure (continued)

12. Modify the shift register circuit so that it conforms to that shown in Figure 7-76. The parallel data inputs will not be used in this step. The mode control input line is simply connected to ground. The output from the D flip-flop is connected through one of the inverters in a 74LS04 IC and then fed back around to the input of the shift register.

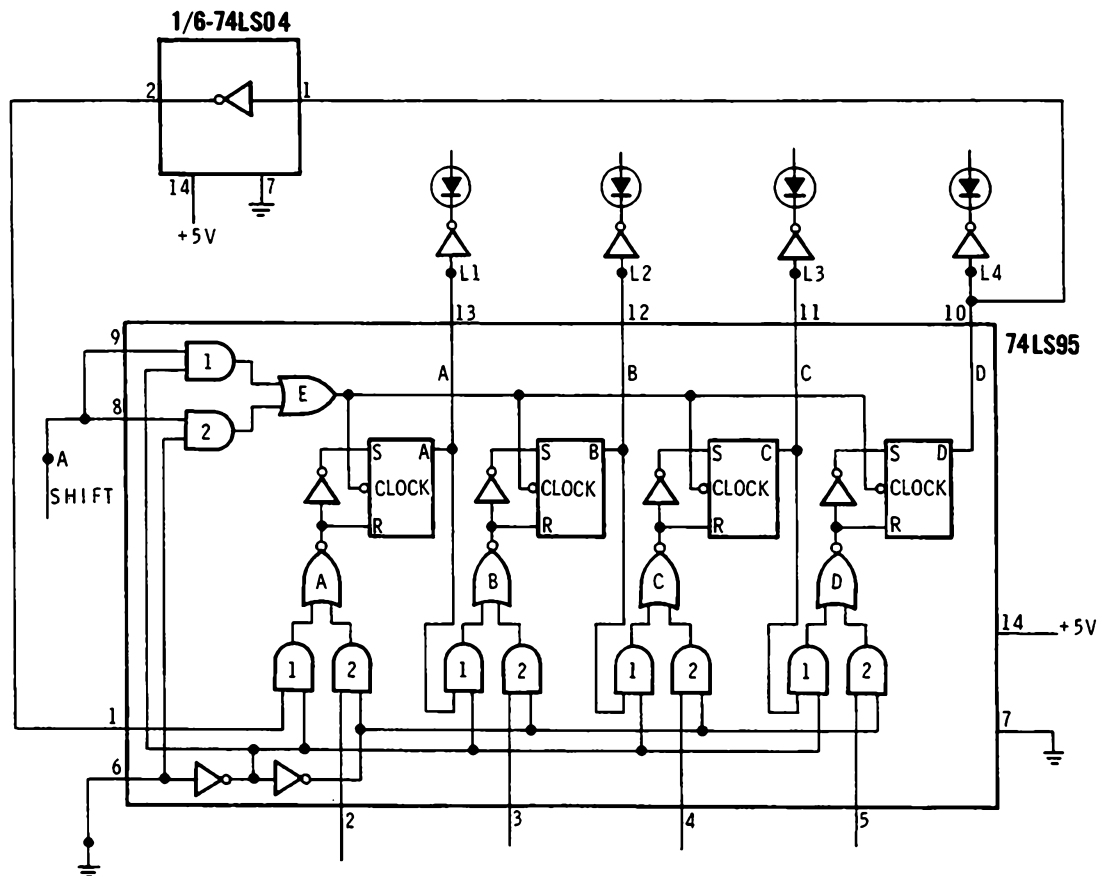


Figure 7-76
Shift register circuit for Steps 12 through 15.

13. Study the circuit in Figure 7-76 and answer the following questions.

The circuit is set up to perform a shift (right, left) _____ operation.

The shift register circuit connected in this way is known as a _____ counter.

14. Apply power to the circuit. Depress the A shift input switch until the shift register contains all 0's. Record this state in the first position of Table III. Next, depress the A logic switch and after each actuation record the shift register state in the sequential locations in Table III. Continue depressing the A logic switch one at a time and recording the register states until Table III is complete.

TABLE III

A	B	C	D

15. Disconnect the shift clock input lines at pins 8 and 9 on the 74LS95 IC from the A logic switch output and connect them to the clock (CLK) output. Set the clock frequency to 1 Hz. Apply power to the circuit and observe the shift register output states. As soon as all the LED indicators show the register content to be 0000, monitor the shift register states after each clock pulse and verify them against the data you collected and recorded in Table III. Continue to let the shift register circuit operate while observing Table III. Be sure that you let the circuit run long enough so that you understand the operation that is taking place.


Discussion of Steps 12 through 15

The circuit you constructed in Step 12 is a Johnson counter. Since the mode control input is set to binary 0 by grounding it, the circuit will perform a shift right operation. The circuit connection feeds the normal output of the D flip-flop through an inverter and applies the complement back around to the serial input of the shift register. This is the equivalent of connecting the normal and complement output of the shift register output flip-flop back around to the K and J inputs of the input flip-flop on a shift register as indicted previously in the unit. When we do this we form a switch tail or Johnson counter. Since the normal and complement flip-flop outputs of the JK inputs are not available, the arrangement in Figure 7-76 accomplishes the same effect. Essentially we invert the output of the shift register and feed it back to the serial input. The result is a four bit Johnson counter. Such a counter has $2N$ discrete states where N is the number of flip-flops. Since we are using four flip-flops this circuit should have 8 discrete states. You verified this by stepping the counter and recording the flip-flop states in Table III. You then verified this operation by letting the counter operate automatically from the 1 Hz clock pulse. You should have found the shift register states to be like those indicated in Table IV.

TABLE IV

A	B	C	D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1

RECYCLE



EXPERIMENT 17

Clocks and One Shots

OBJECTIVES: *To demonstrate the operation of several clock oscillator circuits and a retriggerable one shot multivibrator.*

Materials Required

Heathkit Digital Design Experimenter ET-3200
2 — MPS A20 transistors (417-801)
1 — 150 ohm resistor
2 — 1 k ohm resistors
2 — 4.7 k ohm resistors
2 — 1000 μ F electrolytic capacitors
1 — type 74LS04 TTL IC (443-755)
1 — type 74LS123 TTL IC (443-942)
1 — 47 k ohm resistor

Procedure

1. Construct the astable multivibrator circuit shown in Figure 7-77. Take your time in wiring the circuit to be sure that you do not make any wiring mistakes. You will observe the operation of this circuit on LED indicators L1 and L2.

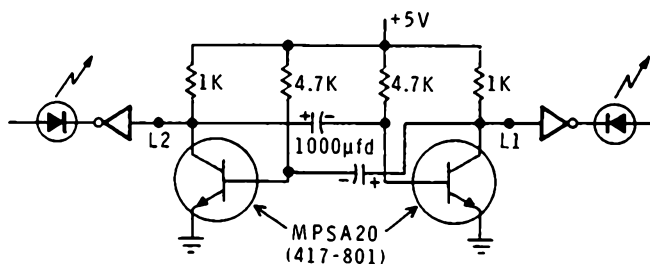


Figure 7-77
Astable multivibrator experimental circuit.

2. Using the component values indicated in Figure 7-77, compute the frequency of oscillation of this astable multivibrator circuit. Record your answer below.

$$F = \text{_____ Hz}$$

Next, compute the period of oscillation for this circuit.

$$\text{Period} = \text{_____ seconds}$$

The duty cycle of this circuit will be _____.

3. Apply power to the circuit and observe LED indicators L1 and L2. Use the sweep-second hand on your watch to measure the period of oscillation. Record your measured value below and compare to the computed value you determined earlier.

$$\text{Period} = \text{_____ seconds}$$

Does the actual operation of the circuit correspond to the answers you gave in Step 2 above? Account for any discrepancy that you might observe.

4. Construct the clock oscillator circuit shown in Figure 7-78. You will use a type 74LS04 TTL hex inverter. Be sure to connect +5 volts and ground to the integrated circuit. You will observe the circuit output on LED indicator L4.
5. Apply power to the circuit. Measure the period of oscillation using the sweep-second hand on your watch and record below.

$$\text{Period} = \text{_____ seconds}$$

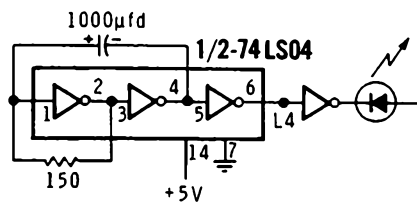


Figure 7-78
IC clock oscillator circuit.

Discussion of Steps 1 through 5

In these steps you demonstrated two types of clock oscillators. The discrete component clock in Figure 7-77 is a standard astable multivibrator. The circuit should switch repeatedly between its two states as indicated by LED indicators L1 and L2. When L1 is on L2 will be off and vice-versa.

Using the formula given earlier in the unit, the frequency of oscillation should be about .154 Hz. This means that the circuit should have a period (time for one cycle) of approximately $1 \div .154$ or about 6.6 seconds. This means that the circuit should change state every 6.6 seconds. Each LED indicator will remain on for approximately 3.25 seconds. This very low frequency of oscillation is caused primarily by the very high value of capacitance used in the circuit. Higher frequencies can be obtained by using smaller capacitor values.

Your measured period may be somewhat different from your calculated value. The formula given for computing the frequency is an approximation to begin with, but the most likely cause for the difference between your computed and measured values is the tolerances of the timing resistors and capacitors. Since both capacitors are the same and the base resistors are equal, the duty cycle of the circuit should be 50 percent.

In Steps 4 and 5 you demonstrated a clock oscillator circuit made from TTL inverters. As indicated earlier in the unit, the period of oscillation of the circuit is approximately 3 RC. Using the component values shown in Figure 7-78, the period of oscillation should be approximately .45 second. This represents a frequency of 2.22 Hz. In other words, the LED indicator L4 should flash on and off once every .45 seconds. Because of the asymmetrical nature of this circuit, the duty cycle will be less than 50 percent.

Procedure (continued)

6. Disassemble the clock oscillator circuits you constructed in the previous steps. On the breadboarding socket of your Trainer, construct the circuit shown in Figure 7-79. This circuit uses the 74LS123 dual retriggerable one-shot. The circuit is wired so that the first one-shot will be triggered from the A logic switch. This one-shot will in turn trigger the second one-shot in the IC. External resistors and capacitors are used to set the duration of the pulses produced by each one-shot. You will monitor the one-shot outputs on LED indicators L1 and L4. Note that the logic switch output, \overline{B} , is connected to the reset (C) input of the first one-shot. Don't forget to connect +5 volts to pin 16 and ground to pin 8 of the IC.

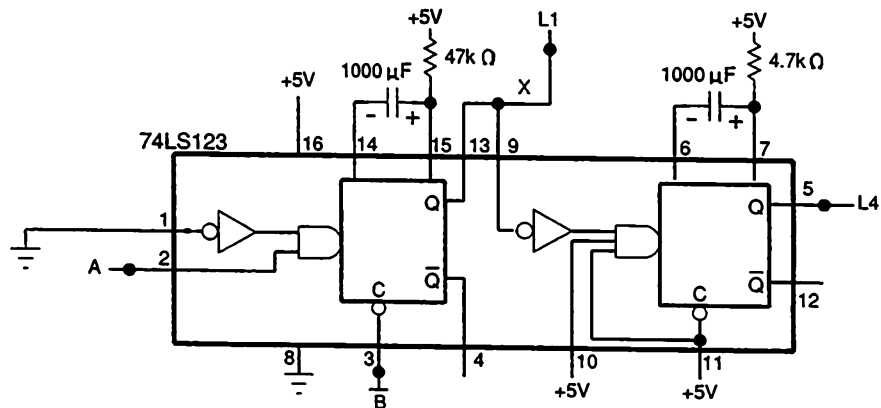


Figure 7-79

One shot experimental circuit.

The pin connections for the 74LS123 IC are shown in Figure 7-80. The pulse duration produced by this one-shot is a function of the external component values R and C and can be computed with the formula below.

$$t = 0.33 RC$$

In this formula the resistance value R is in $k\Omega$ and the capacitance value C is in μF . The output pulse duration t will be in milliseconds.

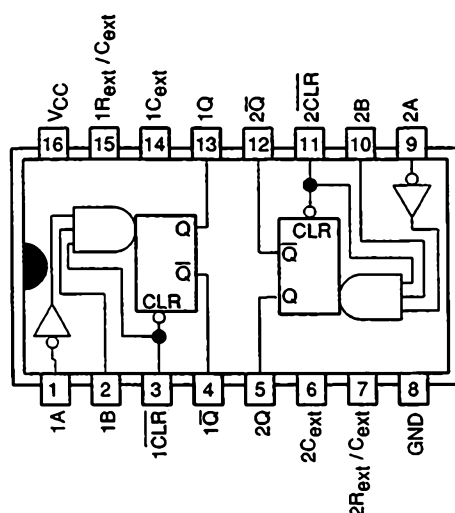


Figure 7-80
Pin connections for
74LS123 dual retriggerable one shot.

7. Using the formula, compute the pulse duration of each one shot in Figure 7-79. Record your pulse durations below. The output pulse width of the first one-shot is t_1 and the output of the second one-shot is t_2 .

$$t_1 = \underline{\hspace{2cm}} \text{ ms}$$

$$t_2 = \underline{\hspace{2cm}} \text{ ms}$$

8. Study the circuit shown in Figure 7-79. Determine the operation of the circuit. Assume that the circuit operation is initiated by actuating the A logic switch. Sketch the input and output waveforms for the circuit.

Will the circuit be triggered when the A logic switch is depressed or released? _____.

9. Depress the A logic switch and release it. Note the LED indicators to see what operation occurs. Use the second hand of your watch to time the one shot output durations by observing L1 and L4. Repeat the sequence as often as necessary to verify the operation of the circuit.

Does the actual operation of the circuit correspond to your result in Step 8? _____.

10. Momentarily depress and release the A logic switch. Note LED indicator L1. After a second or two, depress the B logic switch while noting L1 and L4. What happens? _____.

11. Remove the input from logic switch A to pin 2 of the IC and connect pin 2 to the clock output CLK. Set the clock frequency to 1 Hz and observe the L1 indicator.

What is the state of L1? _____.

What does this state indicate? _____.

Discussion of Steps 6 through 11

In these steps you demonstrated the operation of a retriggerable one-shot. The retriggerable function is not always used, and when it isn't, this circuit performs like any other monostable multivibrator. The circuit you constructed receives a trigger pulse from the A logic switch. The A output normally rests in the low position. When the switch is depressed, A goes high and when it is released goes low again. It is on the low to high transition that the input one-shot is triggered. When it is triggered, LED indicator L1 will turn on. This output will remain on until the pulse duration specified by the external resistor and capacitor is completed. According to the calculations using the formula given earlier, the pulse output duration for this one-shot (t_1) should be 15.5 seconds.

When the input one-shot times out, its output will switch from high to low. This will trigger the second one-shot in the circuit. Its time constant is set to produce an output pulse (t_2) of 1.55 seconds. Therefore, as soon as L1 turns off, L4 should turn on for approximately 1.55 seconds and then go out. This sequence can be repeated by depressing the A logic switch again.

The B logic switch is wired to the clear input of the first one-shot. If you trigger the circuit into operation with the A logic switch, the first one-shot will remain on for about 15.5 seconds. However, this timing interval can be terminated or cut short by applying a reset pulse with logic switch B. The moment the B logic switch is depressed, the one-shot output will switch off. LED L1 will go out. This will immediately trigger the second one-shot and cause L4 to light for approximately 1.55 seconds. Both operations could be terminated by connecting the \bar{B} logic switch output to the clear input of the second one-shot as well.

The one-shot circuit that you demonstrated here shows how a delay function is implemented. The first one-shot produces a delay of over 15 seconds and the second one-shot generates a single output pulse 1.55 seconds long. The A logic switch initiates the circuit operation, but it is the output of the second one-shot that is generally used to actuate an external circuit. See Figure 7-81.

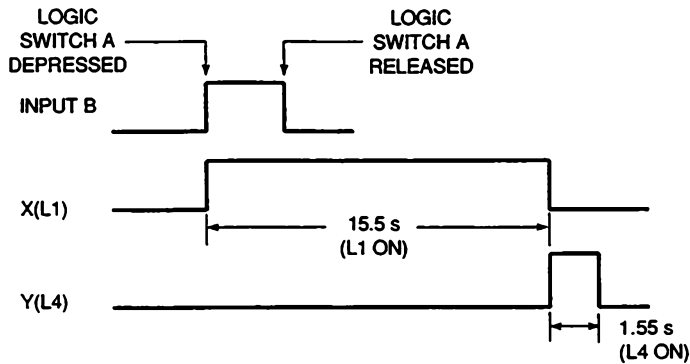


Figure 7-81

Waveforms illustrating the operation of the experimental one shot circuit.

Finally, you connected the input to the circuit to the 1 Hz output of your clock. Since the clock interval is approximately one second, the input one-shot will be repeatedly triggered. This will cause the output of the one-shot to turn on. Before the circuit can time out, the input will be triggered again. Therefore, the output of the first one-shot will remain on as long as the clock signal is applied. When the pulse duration of the one-shot is greater than the period of the input triggering signal, the retriggerable feature comes into operation and will keep LED indicator L1 turned on. LED indicator L4 will remain off during this time since the second one-shot will not be triggered.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and work every problem first before checking the answers.

1. Which of the following is **not** a sequential logic circuit?
 - a. counter
 - b. register
 - c. NOR gate
 - d. one shot
 - e. clock
2. A binary up counter with flip-flops FEDCBA (A = LSB) is preset to the 001101 state. After a number of input pulses are applied, the new counter state is 100011. How many pulses occurred?
 - a. 13
 - b. 22
 - c. 26
 - d. 35
3. A binary counter with 8 flip-flops has a maximum counter capability of:
 - a. 8
 - b. 63
 - c. 127
 - d. 255
4. A binary counter with 12 flip-flops will divide an input frequency by:
 - a. 12
 - b. 4096
 - c. 8192
 - d. 16384
5. Another name for a BCD counter is:
 - a. decade counter
 - b. binary counter
 - c. frequency divider
 - d. shift register

6. Which of the following is an invalid code in a BCD counter?
- 0101
 - 1001
 - 1101
 - 0001
7. How many BCD counters does it take to count to 102557?
- 5
 - 6
 - 12
 - 24
8. Two cascaded BCD counters divide an input frequency by:
- 2
 - 8
 - 100
 - 256
9. The output frequency of the circuit in Figure 7-82 is:
- 120 KHz
 - 240 KHz
 - 1.2 MHz
 - 2.4 MHz

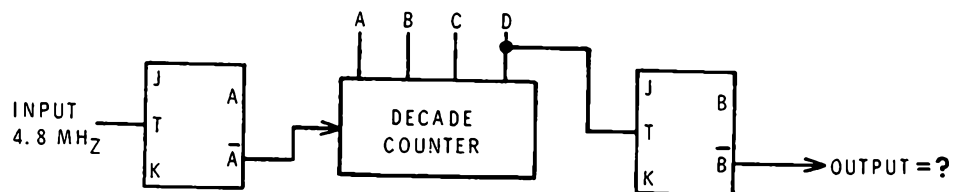


Figure 7-82
Circuit for Question 9.

10. An 8-bit binary up/down counter is initially preset to 10000001. Twenty-seven input pulses increment the counter. Eighty-eight pulses decrement the counter. The binary content then is:
- 10011100
 - 01101001
 - 10010110
 - 01000100

11. Clock pulses are applied simultaneous to all flip-flops in which of the following?
- synchronous binary counter
 - BCD ripple counter
 - shift register
 - one shot
12. How many BCD digits can be stored in an eight bit shift register?
- 1
 - 2
 - 3
 - 4
13. An 8-bit shift right register contains the number 01011010. A binary 0 is applied to the serial input. After three clock pulses occur, the number in the shift register is:
- 11010111
 - 00001011
 - 11010000
 - 11101011
14. Parallel to serial data conversion can be performed by a:
- BCD counter
 - binary counter
 - shift counter
 - shift register
15. An 8-bit shift register contains the number 00001000. Three clock pulses are applied shifting the word to the left. The operation performed is:
- multiplication by 3
 - division by 3
 - multiplication by 8
 - division by 8
16. The single input to a shift register is:
- serial
 - parallel
- and the flip-flop outputs are:
- serial
 - parallel

17. Which of the following is **not** a name for the circuit that generates pulses to operate a sequential circuit?
- a. one shot
 - b. clock
 - c. astable multivibrator
 - d. oscillator
18. The circuit that can delay an input by a fixed duration each time it is triggered is called a(n):
- a. astable multivibrator
 - b. monostable
 - c. clock
 - d. pulser
19. Which of the following requires data recirculation and a continuous minimum frequency clock for proper operation?
- a. synchronous counter
 - b. Johnson counter
 - c. TTL shift register
 - d. dynamic MOS shift register
20. With external R and C values of 10K ohms and .0022 μf , the one shot in Figure 7-80 will generate an output pulse with a duration of:
- a. .005885 μS
 - b. 5.885 μS
 - c. 588.5 μS
 - d. 5.885 mS

EXAMINATION ANSWERS

1. c.—NOR gate is not a sequential circuit.
2. b.—22. The counter is preset to 13. After the input pulses occur, the count is 35. Therefore, $35 - 13 = 22$ pulses occurred.
3. d.—255. $N = 2^n - 1 = 2^8 - 1 = 256 - 1 = 255$
4. b.—4096 $N = 2^n = 2^{12} = 4096$
5. a.—decade counter
6. c.—1101 Numbers between 1010 (10) and 1111 (15) are invalid in BCD code.
7. b.—6 One BCD counter for each decimal digit.
8. c.—100 Each BCD counter divides by 10.
9. a.—120 KHz The input flip-flop divides by 2, the decade counter by 10, and the output flip-flop by 2 for a total division of $2 \times 10 \times 2 = 40$. The output is $4.8 \text{ MHz} \div 40 = .12 \text{ MHz}$, or 120 KHz.
10. d.—01000100 The counter is preset to 129. Twenty-seven pulses increment this to $129 + 27 = 156$. Eighty-eight pulses decrement this to $156 - 88 = 68$ (01000100).
11. a.—synchronous binary counter
c.—shift register
12. b.—2 Each BCD digit has four bits.
13. b.—00001011 The shift sequence is:

01011010	
00101101	0
00010110	10
00001011	010
14. d.—shift register
15. c.—multiplication by 8
16. a.—serial
b.—parallel
17. a.—one shot All the others are some form of clock oscillator.
18. b.—monostable (or one shot)
19. d.—dynamic MOS shift register
20. b.—5.885 μS

$$t = .25 RC \left(1 + \frac{.7}{R}\right) = .25 (10)(.0022)(1.07)$$

$$t = .005885 \text{ milliseconds (mS) or}$$

$$5.885 \text{ microseconds } (\mu\text{S})$$

Unit 8

**COMBINATIONAL
LOGIC CIRCUITS**

CONTENTS

Introduction	8-3
Unit Objectives	8-4
Unit Activity Guide	8-5
Decoders	8-6
Encoders	8-17
Multiplexers	8-22
Demultiplexers	8-34
Exclusive OR	8-38
Code Converters	8-53
Read Only Memories	8-57
Programmable Logic Arrays	8-75
Experiment 18 — Decoders	8-79
Experiment 19 — 7 Segment Decoder-Driver and Display . . .	8-85
Experiment 20 — Multiplexers	8-93
Experiment 21 — Exclusive OR	8-101
Experiment 22 — Exclusive OR Applications	8-107
Unit Examination	8-119
Examination Answers	8-123

INTRODUCTION

Combinational logic circuits are digital circuits that are made up of gates and inverters. The output of a combinational logic circuit is a function of the states of its inputs, the types of gates used, and how they are interconnected. As you saw in a previous unit on Boolean algebra, there are many different ways to interconnect logic gates to form combinational circuits. Any unique binary function can be implemented.

An analysis of a wide variety of different types of digital equipment reveals that there are certain combinational logic circuits that regularly reoccur. Despite the large possible number of combinational circuits, most digital equipment can be implemented with just a few basic types. These circuits are called functional logic circuits. The most common functional logic circuits are decoders, encoders, multiplexers, comparators, and code converters.

In this unit, you are going to study the most common types of functional combinational logic circuits. You will learn how they operate and how they are used. You will see that even though you can construct these common functional circuits from gates and inverters, in most cases these functional logic circuits are already available as a completely wired and ready to use MSI integrated circuit. The availability of these functional circuits in MSI form usually eliminates the need to design them. In designing digital equipment, you will find that the job is largely one of identifying the functional circuits, selecting appropriate MSI devices, and interconnecting them properly.

The Unit Objectives outline specifically what you will learn in this unit. Review these now, then go on to the Unit Activity Guide for your specific instructions in completing this unit.

UNIT OBJECTIVES

When you complete this unit you will be able to:

1. Name at least seven different types of standard combinational or functional logic circuits.
2. Write the output states of a decoder, encoder, multiplexer, demultiplexer, given the input states.
3. Implement a decoder for any states with NAND or NOR gates.
4. Name two applications for a multiplexer circuit.
5. Write the output states of an exclusive OR and an exclusive NOR circuit given the input states.
6. List three applications for the exclusive OR gate.
7. Explain the operation of a read only memory.
8. Give three applications for a ROM.
9. Define a programmable logic array.

UNIT ACTIVITY GUIDE

**Completion
Time**

<input type="checkbox"/> Read "Decoders".	_____
<input type="checkbox"/> Answer Self Test Review questions 1-6	_____
<input type="checkbox"/> Read "Encoders".	_____
<input type="checkbox"/> Answer Self Test Review questions 7-10	_____
<input type="checkbox"/> Read "Multiplexers".	_____
<input type="checkbox"/> Answer Self Test Review questions 11-15	_____
<input type="checkbox"/> Read "Demultiplexers".	_____
<input type="checkbox"/> Answer Self Test Review questions 16-19	_____
<input type="checkbox"/> Read "Exclusive OR".	_____
<input type="checkbox"/> Answer Self Test Review questions 20-27	_____
<input type="checkbox"/> Read "Code Converters".	_____
<input type="checkbox"/> Answer Self Test Review questions 28-31	_____
<input type="checkbox"/> Read "Read Only Memories".	_____
<input type="checkbox"/> Answer Self Test Review questions 32-45	_____
<input type="checkbox"/> Read "Programmable Logic Arrays".	_____
<input type="checkbox"/> Answer Self Test Review questions 46-50	_____
<input type="checkbox"/> Perform Experiment 18.	_____
<input type="checkbox"/> Perform Experiment 19.	_____
<input type="checkbox"/> Perform Experiment 20.	_____
<input type="checkbox"/> Perform Experiment 21.	_____
<input type="checkbox"/> Perform Experiment 22.	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Review the Examination Answers.	_____

DECODERS

One of the most frequently used combinational logic circuits is the decoder. A decoder is a logic circuit that will detect the presence of a specific binary number or word. The input to the decoder is a parallel binary number and the output is a binary signal that indicates the presence or absence of that specific number.

The basic decoding circuit is an AND gate. The output of an AND gate is a binary 1 only if all inputs are a binary 1. By properly connecting the inputs on an AND gate to the source of the data, the presence of any binary number will be detected.

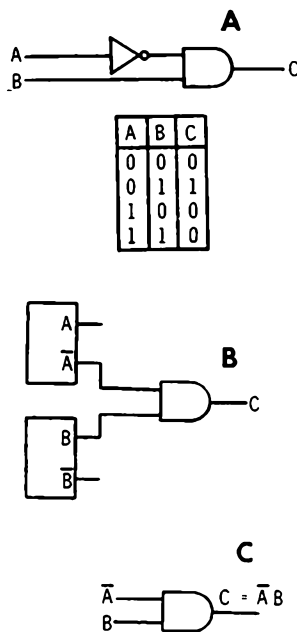


Figure 8-1
Two input AND gate
decoders used for detecting
the number 01.

Figure 8-1A shows a two input AND gate used to detect the presence of the two bit binary number 01. The number to be detected consists of two bits, A and B, with B the least significant bit (LSB). When A is 0 and B is 1, both inputs to the AND gate will be high and the output C will be a binary 1 indicating the presence of the desired number. The inverter on the A input causes the upper input to the AND gate to be binary 1 when the A input is binary 0. For any other combination of input bits the decoder output will be binary 0.

The truth table accompanying the circuit in Figure 8-1A illustrates the performance of the circuit. Note that when the input number is 01 the output C is binary 1. For all other two input combinations the output is binary 0. The circuit does indeed detect the presence of the number 01.

Figure 8-1B shows the AND gate decoder for detecting the number 01 where the binary number input source is a flip-flop register. Since the complement outputs of the flip-flops are available, the inverter is not needed. When the A flip-flop is reset and the B flip-flop is set, the number stored in the register is 01. At this time the \bar{A} and B outputs are high. The decode gate output will be high at this time.

To simplify the drawing of a decoder circuit, the AND gate input source is often omitted. See Figure 8-1C. Only the input states are shown at the gate inputs. Note the output equation which can be written from the circuit or the truth table.

An AND gate can be used to detect the presence of any binary number regardless of size. The number of inputs to the gate will be equal to the number of bits in the binary word. Figure 8-2 shows how a four input gate can be used to detect the binary number ABCD = 0101. Note that the decode gate receives its inputs from a 4 bit register. When the number 0101

is present in the register the output of the decode gate will be a binary 1. For any other 4 bit number in the register, the decoder output will be a binary 0.

While there are some situations where the presence of a single binary word must be recognized, most applications require the detection of all possible states that can be represented by the input word. For example, with a two bit input word there are a total of $2^2 = 4$ different input combinations that exist. A practical decoder will recognize the existence of each of these states.

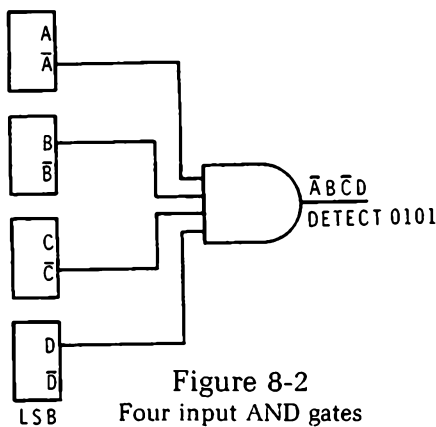
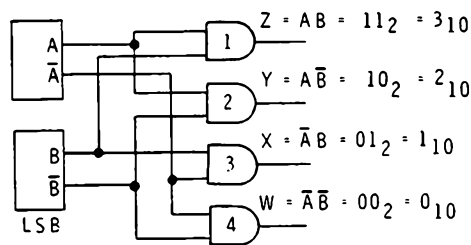


Figure 8-2
Four input AND gates
used to decode 0101.



INPUTS		OUTPUTS			
A	B	W	X	Y	Z
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Figure 8-3
1 of 4 decoder.

Figure 8-3 shows such a decoder. A two bit binary word with bits A and B (B is the LSB) is stored in a flip-flop register. Four AND gates are used to decode the four possible combinations. For example, gate 4 detects the 00 input state. If the binary number stored in the flip-flops is 00, the \bar{A} and \bar{B} outputs will be high. Gate 4 will produce a binary 1 output. Gates 1, 2, and 3 will have a binary 0 on at least one of their inputs thereby keeping their outputs low. The truth table in Figure 8-3 shows the four possible input states and the outputs of each of the decoder gates.

Another way of looking at the decoder circuit in Figure 8-3 is as a binary to decimal converter. It converts a binary number into an output signal representing one of the four decimal numbers 0, 1, 2, or 3. If flip-flops A and B are both set, the register is storing the binary number 11_2 . Gate 1 will be enabled at this time and its output will indicate the presence of that particular number in the register. The output of this gate could then be used to turn on an indicator light marked with the decimal number 3.

BCD to Decimal Decoder

One of the most common applications for decoder circuits is binary to decimal conversion. A widely used type of decoder is the BCD to decimal decoder. The input to the decoder is a parallel four bit number representing the BCD digits 0000 through 1001. Ten NAND gates are used to look at or observe the inputs and decode the ten possible output states 0 through 9. When a BCD number is applied to this decoder, one of the ten output lines will go low indicating the presence of that particular BCD number. The output of such a decoder is generally used to operate a lighted decimal number read-out or display.

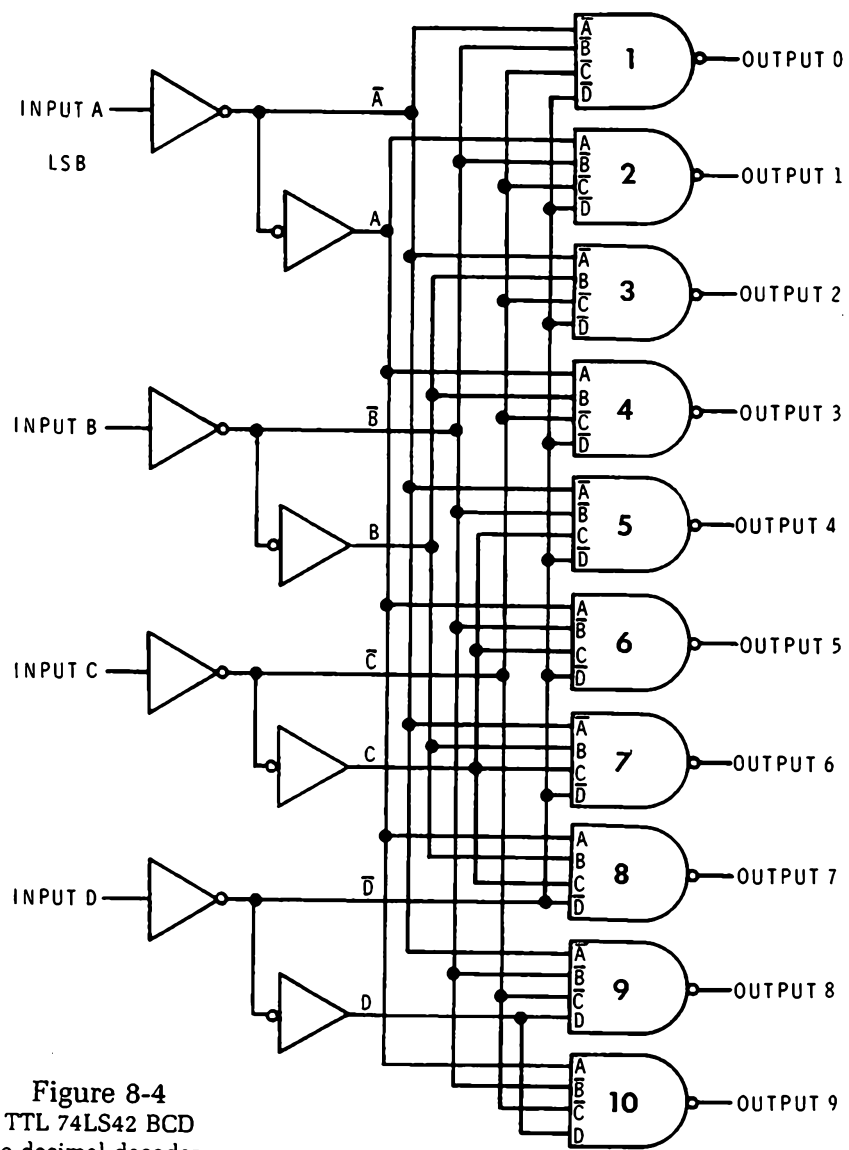


Figure 8-4
TTL 74LS42 BCD
to decimal decoder.

A typical BCD to decimal decoder is shown in Figure 8-4. The four bit BCD number with bits designated A, B, C, and D are applied to the inverters which generate the normal and complement versions of the inputs to be applied to the decode gates. Bit A is the LSB. Note also that NAND gates are used instead of AND gates for the decoding process. When all of the inputs to a NAND gate are binary 1, its output will be binary 0. For all other input combinations the output will be binary 1. For that reason, all of the outputs from the gates in this decoder are high except the one decoding a specific input state.

Figure 8-5 shows the truth table for the BCD to decimal decoder. When one of the ten 8421 BCD codes is applied to the input, the appropriate output will go low. For example, when the input 0110 is applied to the decoder, all inputs to gate 7 will be binary 1. The output of gate 7 will go low indicating the presence of the four bit BCD number representing a decimal 6 is at the input. All other gate outputs will be high at this time. Notice in the truth table that if any one of the six invalid four bit BCD code numbers is applied to the input of the decoder, all outputs will remain high. This BCD decoder simply does not recognize the four bit words that are not included in the standard 8421 BCD code.

NO.	BCD INPUT				DECIMAL OUTPUT									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
INVALID	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = BINARY 1

L = BINARY 0

Figure 8-5
Truth table for BCD
to decimal decoder.

The decoder circuit in Figure 8-4 can be readily constructed with individual logic gates. A typical SSI logic gate provides two four input NAND gates in a single dual in line package. To decode the 10 output states, five of these integrated circuits would be required. The inverters could be implemented with a hex inverter IC. A typical unit contains six inverter circuits in a single DIP. Since 8 inverters are required, two of these hex inverter ICs would be required. This makes a total of seven integrated circuits required to implement the BCD to decimal decoder. Some form of printed circuit board or other interconnecting medium would be required to wire the circuit as indicated.

Fortunately, modern integrated circuit technology has eliminated the necessity for constructing such a circuit with SSI logic circuits. The entire BCD to decimal decoder circuit shown in Figure 8-4 is available in a single 16 pin dual in line package. Because of the complexity of this circuit it is considered to be a medium scale integrated circuit. This is a classical example of a functional MSI logic circuit.

Octal and Hex Decoders

Octal and hexadecimal decoder circuits are also widely used. The octal decoder accepts a parallel three-bit input word and decodes all eight output states representing the numbers 0 through 7. The BCD to decimal decoder circuit in Figure 8-4 can be used as an octal decoder by simply using the A, B, and C inputs only. The D input is simply wired to a binary 0 condition and the 8 and 9 outputs from gates 9 and 10 are ignored. A hex decoder decodes all 16 states represented by four input bits.

Instead of drawing the complex logic circuitry for a decoder, a simplified block diagram like the one shown in Figure 8-6 is often used. The decimal weights of the inputs and the decimal equivalent of the decoded outputs are indicated within the block to identify its function.

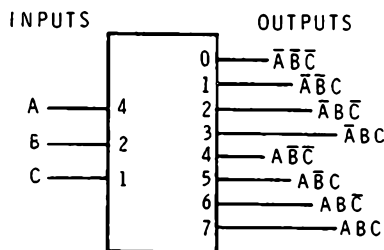


Figure 8-6
Octal decoder.

BCD to 7 Segment Decoder

A special form of decoder circuit is the popular BCD to 7 segment decoder-driver. This is a combinational logic circuit that accepts the standard 8421 BCD input code and generates a special 7 bit output code that is used to operate the widely used 7 segment decimal read-out display. This functional circuit is available as a single package MSI device.

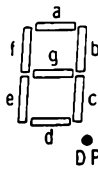


Figure 8-7
7 segment display format.

A 7 segment readout is an electronic component used to display the decimal numbers 0 through 9, and occasionally special letter combinations, by illuminating two or more segments in a specially arranged 7 segment pattern. The standard 7 segment display configuration is shown in Figure 8-7. The segments themselves can be constructed with a variety of light emitting elements such as, an incandescent filament, a light emitting diode, fluorescent tube, gas discharge tube, or a liquid crystal segment. By illuminating the appropriate segments, the numbers 0 through 9 and many letters can be displayed. Figure 8-8 shows the typical 7 segment presentation of these numbers.

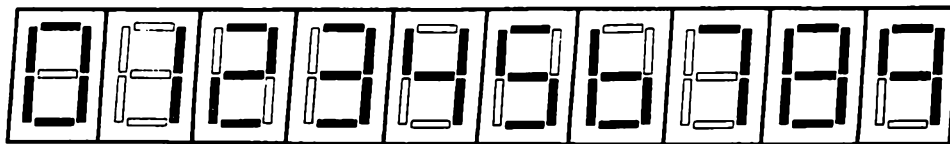


Figure 8-8
7 segment format for
numbers 0 through 9.

Seven segment displays are widely used in electronic equipment such as test instruments, electronic calculators, and digital clocks. Several examples are shown in Figure 8-9. The weather computer in Figure 8-9A uses light emitting diode displays. A light emitting diode is a special semiconductor junction diode that emits light when it is forward biased. Most LED displays emit a brilliant red light. Yellow and green LED displays are also available.



Figure 8-9
Types of 7-segment displays
used in digital equipment: (A) LED.
(B) Liquid Crystal. (C) fluorescent.

The digital multimeter in Figure 8-9B uses a liquid crystal display. The digital clock in Figure 8-9C uses a 7-segment fluorescent display.

These are only a few of the many different types of 7-segment display devices available. A BCD to 7-segment decoder-driver circuit is used to operate these display devices. This is an MSI logic circuit that decodes the decimal states 0 through 9 and develops the seven output signals that operate the display segments.

Figure 8-10 shows the truth table for this decoder circuit. The BCD inputs (ABCD) are in the standard 8421 code form. The outputs are designated a, b, c, d, e, f, g, and correspond to the elements shown in Figure 8-7. A binary 0 in the segment output columns indicates that the corresponding segment is illuminated. You can check this code against the segment letters illustrated in Figure 8-7.

INPUTS					SEGMENT OUTPUTS						
DECIMAL	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	1	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	1	1	0	0

Figure 8-10
Truth table for BCD to
seven segment decoder driver.

A logic diagram for one particular type of BCD to 7 segment decoder-driver is shown in Figure 8-11. In addition to the four BCD inputs, this circuit also has a blanking input, a ripple blanking input, and a lamp test input. When the lamp test input is binary 0 all seven segments of the display are turned on in order to see that none have failed. When the blanking input is low, all segments are turned off. This feature is used where a number of displays are grouped to readout a multi-digit number. This feature blanks or suppresses all leading zeros automatically. For example, in an eight digit display without leading zero suppression, the number 1259 would be displayed as 00001259. With leading zero suppression, only the desired digits 1259 will show. The other four displays will be automatically blanked. By applying a variable duty cycle pulse signal to the ripple blanking input, the intensity of the display can be varied without resorting to supply voltage or current controls.

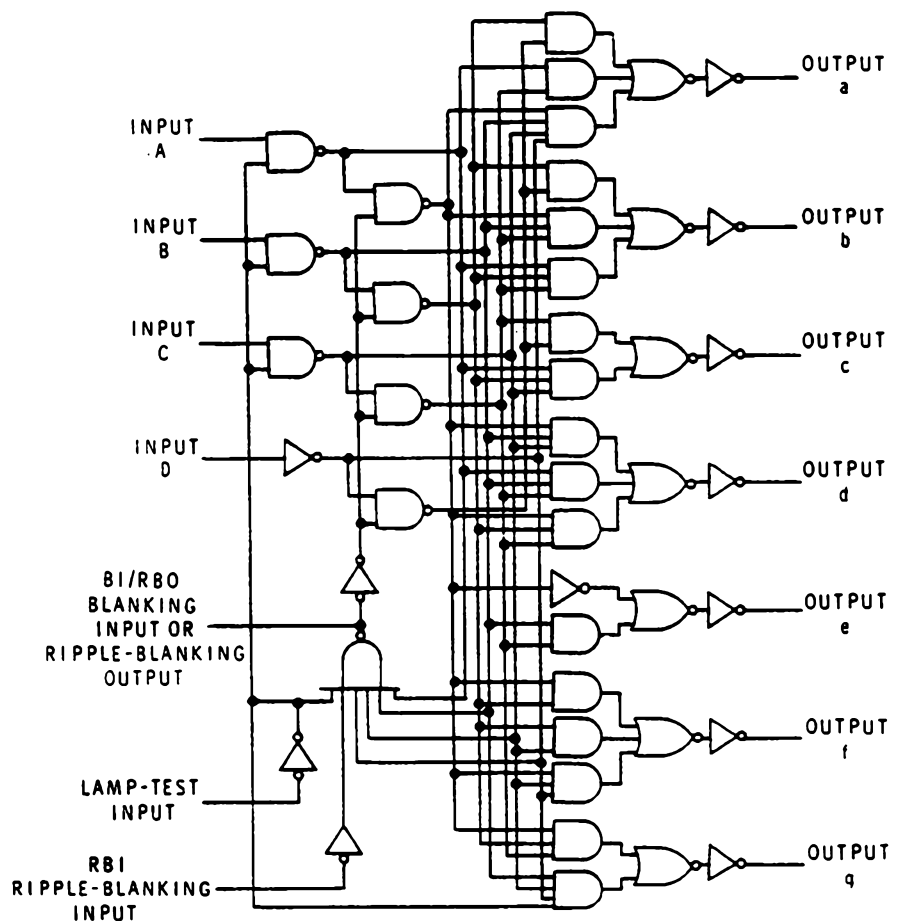


Figure 8-11
BCD-to-seven segment
decoder-driver IC.

Figure 8-12 shows the typical output circuit for one segment of the decoder-driver. Besides decoding the BCD input states this circuit also drives or operates the light producing segments. The output is usually a saturated transistor switch with an open collector. Figure 8-12 shows one LED segment connected to the output. When the transistor conducts, the collector output goes low and current flows through the LED turning it on. A series dropping resistor sets the LED current and hence the intensity of the light it produces.

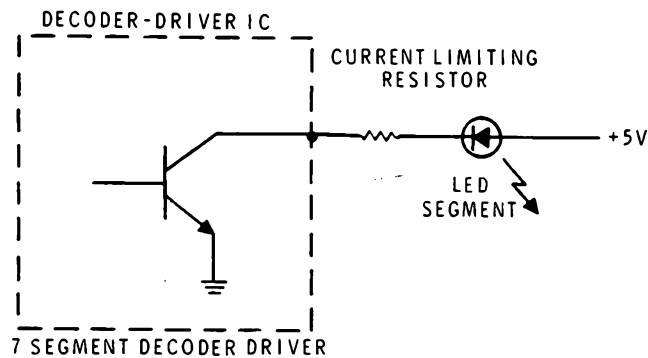


Figure 8-12
Typical 7-segment decoder-driver
output circuit and external connections.

Self Test Review

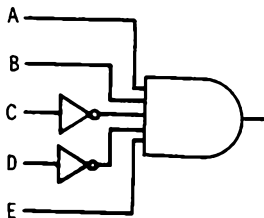


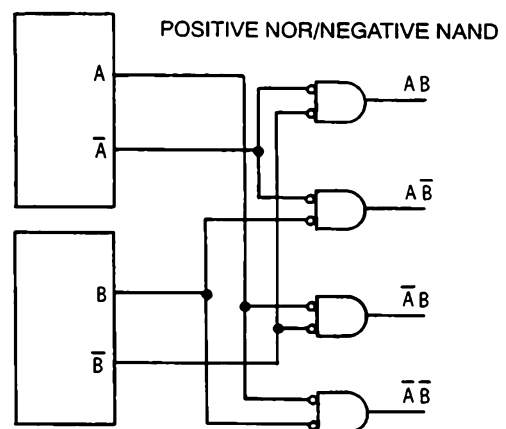
Figure 8-13
Circuit for Self Test
Review question 2.

1. The basic decoder circuit is a(n) _____.
2. What is the decimal equivalent of the state being decoded by the circuit in Figure 8-13? (A is the LSB.) _____
3. In the 7442 BCD-to-decimal decoder (Figure 8-4), the output of gate _____ goes to binary _____ when the input DCBA = 0101.
4. The maximum number of outputs from a decoder with a five-bit input word is _____.
5. Draw a 1-of-4 decoder using 2-input positive NOR/negative NAND gates. Assume that both normal and complement signals are available from two flip-flops, A and B (LSB).
6. Only one output of a 7442 decoder is low while all others are high.
 - a. True
 - b. False

Answers

1. AND gate
2. EDCBA = $10011_2 = 19_{10}$
3. gate 6, binary 0
4. $2^5 = 32$
5. See Figure 8-14
6. a. True

Figure 8-14
Answer to Self Test
Review question 5.

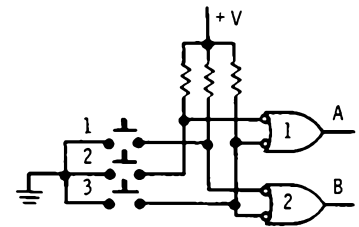


ENCODERS

An encoder is a combinational logic circuit that accepts one or more—inputs and generates a multi-bit binary output code. In a sense, encoders are exactly the opposite of decoders. Decoders detect or identify specific codes while encoders generate specific codes.

Figure 8-15 shows a simple encoder circuit. The inputs are three pushbuttons labeled 1, 2, and 3. The encoder circuit consists of two positive NAND/negative NOR gates. Outputs AB form a two-bit binary code. When pushbutton 1 is depressed, the output of gate 2 goes high. At this time, both inputs to gate 1 are high, therefore, its output is low. By depressing button 1, the output code 01 is generated.

Depressing the number 2 pushbutton forces output A of gate 1 high. The B output of gate 2 is low, therefore, the output code is 10. Depressing button 3 forces the outputs of both gates high, generating the code 11. As you can see, the binary code corresponding to the decimal number given to each input switch is generated when that switch is closed. The truth table in Figure 8-15 summarizes the operation of the circuit. When all of the switches are open (not depressed) the output code is 00.



INPUTS	OUTPUT	
	A	B
1	0	1
2	1	0
3	1	1

Figure 8-15
Simple encoder circuit.

A typical application for an encoder circuit is in translating a decimal keyboard input signal into a binary or BCD output code. Figure 8-16 shows a decimal-to-BCD encoder circuit. When any one of the input lines is brought low, the corresponding 4 bit BCD output code is generated. For example, bringing the number 5 input line low with a pushbutton forces the outputs of gates 1 and 3 high. Gates 2 and 4 have low outputs at this time. As a result, the output code on lines DCBA are 0101, or the binary equivalent of the decimal number 5. This circuit, like all encoders, generates a unique output code for each individual input.

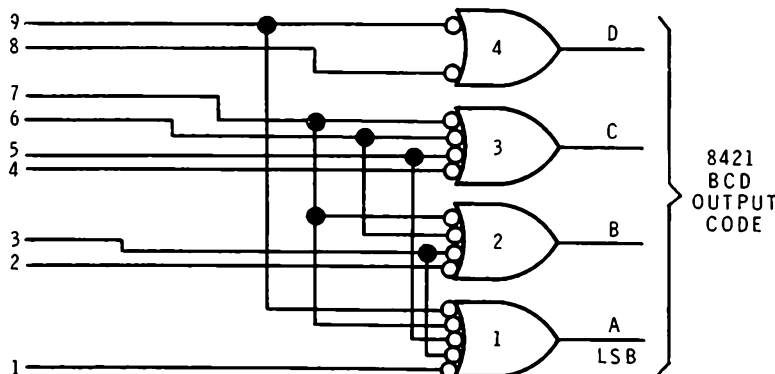


Figure 8-16
Decimal to BCD encoder.

A typical example of a modern integrated circuit binary encoder circuit is shown in Figure 8-17. This is a TTL MSI 8 input priority encoder. The encoder accepts data from 8 input lines and generates the binary code corresponding to the number assigned to the input. The input must be brought low in order to generate the corresponding output code. We say that the inputs are active low. Unlike the two previously discussed encoder circuits, the outputs of the circuit in Figure 8-17 (labeled $\overline{A0}$, $\overline{A1}$, and $\overline{A2}$) are also active low. For this circuit a low output represents a binary 1. This circuit generates a negative logic output code.

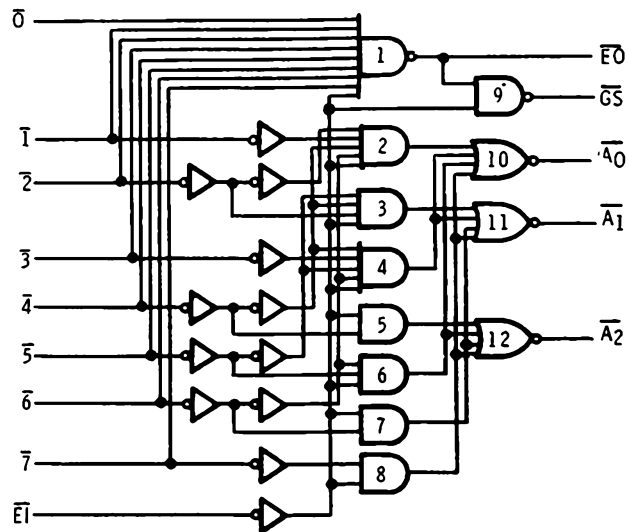


Figure 8-17
Eight input priority encoder circuit with
an \overline{EI} enable input.

A unique feature of this particular circuit is that a priority is assigned to each input so that when two or more inputs are low simultaneously, the input with the highest priority is represented at the output. In this case the inputs with the higher numerical value have the highest priority. This means that if the 3 and 6 inputs are low simultaneously, the binary code representing 6 will be generated at the output.

The \overline{EI} input on this circuit is an enabling input. When this input is high, the output of the inverter connected to it is low. This inhibits gates 1 through 8 and forces the three binary output lines high. When the \overline{EI} input line goes low, the output of the inverter goes high thereby enabling all of the circuitry.

The 9 input NAND gate number 1 monitors all 9 input lines. If any one of them should go low, the \overline{EO} output goes high indicating that one or more of the input lines has been activated. The \overline{GS} output also goes low. If all inputs are high or open (not activated), the \overline{EO} output line is low indicating this state. By using the \overline{EI} input and \overline{EO} and \overline{GS} outputs, several of these devices may be combined to encode N different input states. A truth table for this circuit is shown in Figure 8-18.

TRUTH TABLE

INPUTS									OUTPUTS				
\overline{EI}	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	\overline{GS}	$\overline{A_0}$	$\overline{A_1}$	$\overline{A_2}$	\overline{EO}
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	H	L	L	H
L	X	X	X	X	X	L	H	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	L	L	L	H	H
L	X	X	L	H	H	H	H	H	L	H	L	H	H
L	X	L	H	H	H	H	H	H	L	L	H	H	H
L	L	H	H	H	H	H	H	H	L	H	H	H	H

H=HIGH VOLTAGE LEVEL
L=LOW VOLTAGE LEVEL
X=DON'T CARE (EITHER 1 OR 0)

Figure 8-18
Truth table for nine
input priority encoder.

Self Test Review

7. If inputs 2 and 4 on the encoder in Figure 8-16 are brought low at the same time, the output code will be:
- 0010
 - 0100
 - 0110
 - 1001
8. Answer question 7, above, for the circuit in Figure 8-17.
9. Draw the logic diagram of an encoder to generate the 3-bit binary Gray code given in the table below. Use positive NAND/negative NOR circuits.

INPUT	OUTPUTS		
	A	B	C
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

10. If the 2, 4, and 5 input lines of the priority encoder of Figure 8-17 are brought low simultaneously, the outputs will be:
- $A_0 = H, A_1 = L, A_2 = H$
 - $A_0 = L, A_1 = H, A_2 = H$
 - $A_0 = L, A_1 = H, A_2 = L$
 - $A_0 = L, A_1 = L, A_2 = L$

Answers

7. c. 0110 The outputs of gates 2 and 3 will go high when inputs 2 and 4 go low. In this circuit when two or more inputs are activated at the same time, the output code will be the codes produced by each input alone ORed together.
8. b. 0100 The higher numbered input has priority over the lower numbered input. Only the proper higher numbered output code is generated. This is the reason for the name priority encoder.
9. See Figure 8-19. (Negative logic — Low = 1)

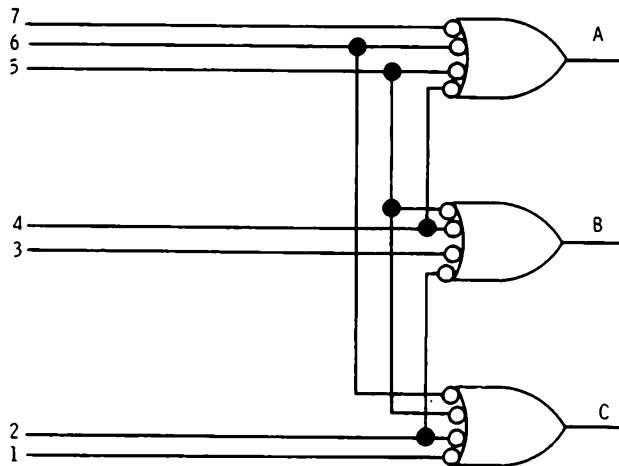


Figure 8-19
Gray code encoder answer
to Self Test Review question 9.

10. c. $A_0 = L, A_1 = H, A_2 = L$
Since $H = 0$ and $L = 1$, the output code will be 101 or 5 which has the highest priority of the three inputs.

MULTIPLEXERS

A multiplexer is an electronic circuit that is used to select and route any one of a number of input signals to a single output. The simplest form of a multiplexer is a single pole multi-position switch. Figure 8-20 shows a rotary selector switch used as a multiplexer. Any one of six input signals can be connected to the output line by simply adjusting the position of this mechanical selector switch. Mechanical selector switches are widely used for a variety of multiplexing operations in electronic circuits. However, many applications require the multiplexer to operate at high speeds and be automatically selectable. Multiplexers of this type can be readily constructed with electronic components.

There are two basic types of electronic multiplexer circuits: analog and digital. The simple selector switch multiplexer in Figure 8-20 will work with either analog or digital signals. However, when electronic multiplexers are constructed, they are primarily designed for either analog or digital applications. For analog applications relays and bipolar or MOSFET switches are widely used. For digital applications involving binary signals, a multiplexer can be simply constructed with standard logic gates. Our primary concern here of course is the digital multiplexer or binary data selector.

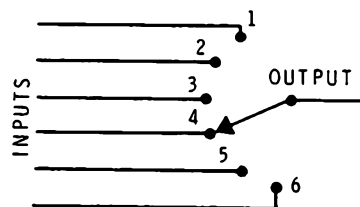


Figure 8-20
A rotary selector switch
used as a multiplexer.

The circuit in Figure 8-21A is the simplest form of digital multiplexer. It has two input data sources and a single output. Either one of the input sources may be selected and fed to the output. The selection process takes place in AND gates 1 and 2. The flip-flop controls these two gates to determine which input is allowed to pass through OR gate 3 to the output. When this flip-flop is set, the Q output will be high enabling gate 1. The \bar{Q} output will be low inhibiting gate 2. Data source 1 will therefore be allowed to pass through gate 1 and through the OR gate 3 to the output. Data source 2 will have no effect on the output state. Resetting the flip-flop reverses this condition. Gate 1 will be inhibited by Q thereby preventing data source 1 from affecting the output. However, data source 2 will be allowed to pass through gates 2 and 3 to the output. This circuit is equivalent to a single pole double throw switch as indicated in figure 8-21 B.

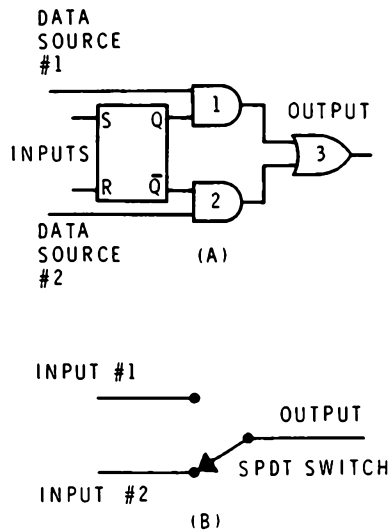


Figure 8-21
Two-input digital
multiplexer (A) and its mechanical
equivalent (B). A SPDT switch.

An MSI functional circuit using this basic two input multiplexer is shown in Figure 8-22. Four 2-input multiplexers are combined to form a multiplexer for two four bit words. Word 1 has bits A1, B1, C1, and D1. Word 2 has bits A2, B2, C2, and D2. The enable (E) input controls the circuit. If E is high, the output of inverter 15 is low thereby inhibiting all of the AND gates and thus preventing either input word from appearing at the outputs. With E low, the circuit is enabled.

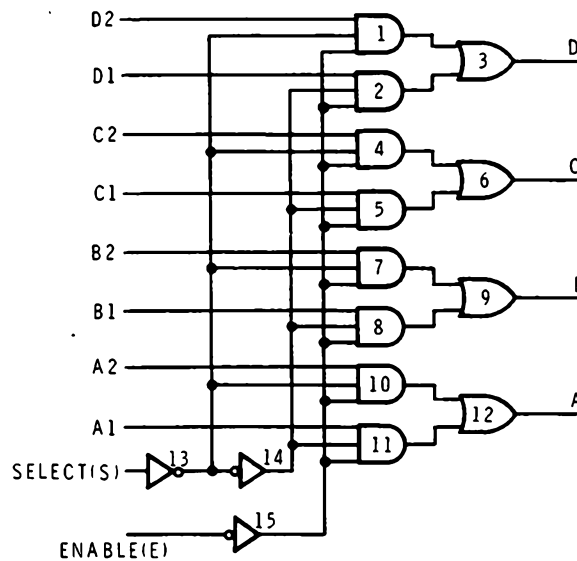


Figure 8-22
Quad two input multiplexer.

The select (S) input specifies which four bit input word appears at the output. When the select input is high, gates 2, 5, 8, and 11 will be enabled letting input word 1 appear at the output. If the S input is low, gates 1, 4, 7, and 10 will be enabled. This permits word 2 to be passed through to the output.

A four input multiplexer circuit is shown in Figure 8-23. Each input is applied to a NAND gate that is enabled or inhibited by a 1 of 4 decoder. The outputs of the NAND gates are ORed together in gate 5. As in other multiplexers, only one of the four inputs will be enabled and allowed to pass through to the output. The selection of the input is made by the decoder circuit. A two bit binary word AB is applied to the decoder. The decoder recognizes one of the four possible input codes and enables the appropriate gate. For example, when the two bit input word is 00 the $\bar{A}\bar{B}$ output line is high. This enables gate 1 and input 1 is allowed to pass through to the output. Input code 01 enables gate 2, input code 10 enables gate 3, and input code 11 enables gate 4.

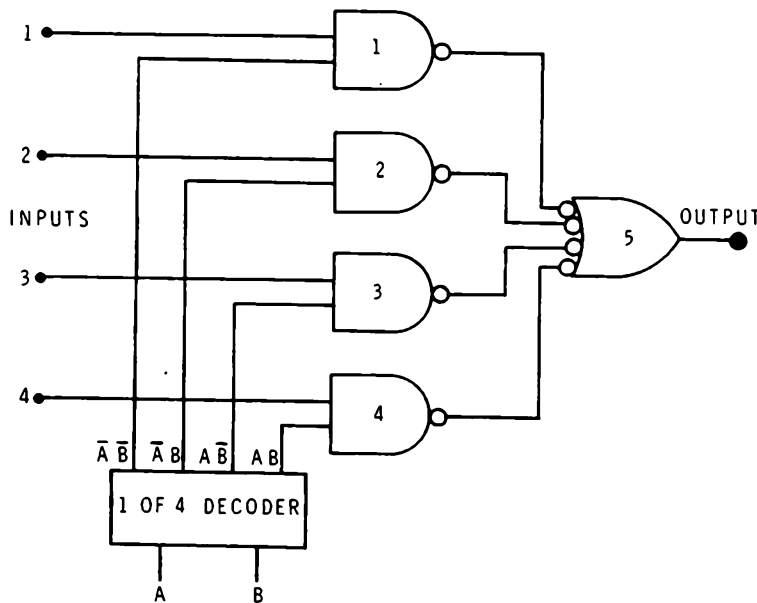


Figure 8-23
4-input multiplexer.

The simplest way to implement the 4 bit multiplexer is to combine both the decode and enable functions in the same gate. Such a circuit is shown in Figure 8-24. The arrangement is virtually identical to the 4 input multiplexer just discussed. However, additional inputs have been added to the input gates so that they also perform the decoding functions. The normal and complement outputs from the two bit binary input word AB are applied to the enable gates in the same way they would be applied to the decoder gates. If the binary input code 00 is applied, the \bar{A} and \bar{B} lines will be high. Gate 1 will be enabled and input number 1 will pass through gate 1 and gate 5 to the output. Gates 2, 3, and 4 will be inhibited at this time.

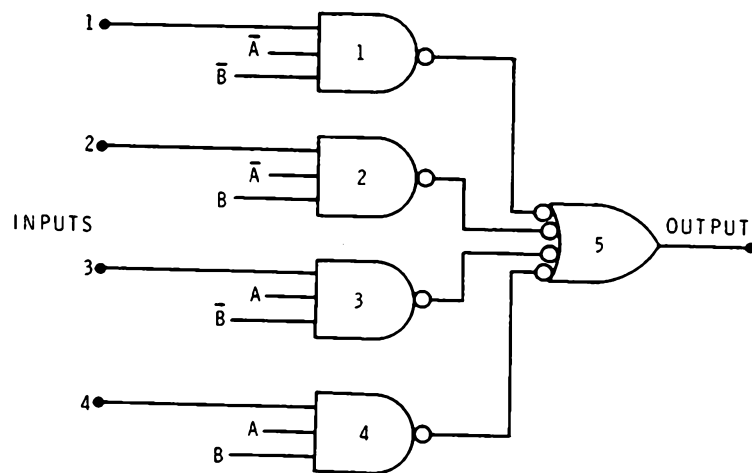


Figure 8-24
A 4-input multiplexer combining
the decode and enable functions.

An eight input TTL binary multiplexer using this same technique is shown in Figure 8-25. Gates 1 through 8 enable or inhibit the eight data input lines D0 through D7. A three bit binary input word (ABC) enables one of the eight gates depending upon the input code. The six inverters at the data select inputs generate the normal and complement signals needed by the select gates. This three input word is an address code that designates which data input line is selected. If the binary input is 101, data input D5 is selected. The strobe enable line enables or inhibits all eight select gates. Both the normal (W) and complement (Y) output signals are available. Another name for the multiplexer is data selector.

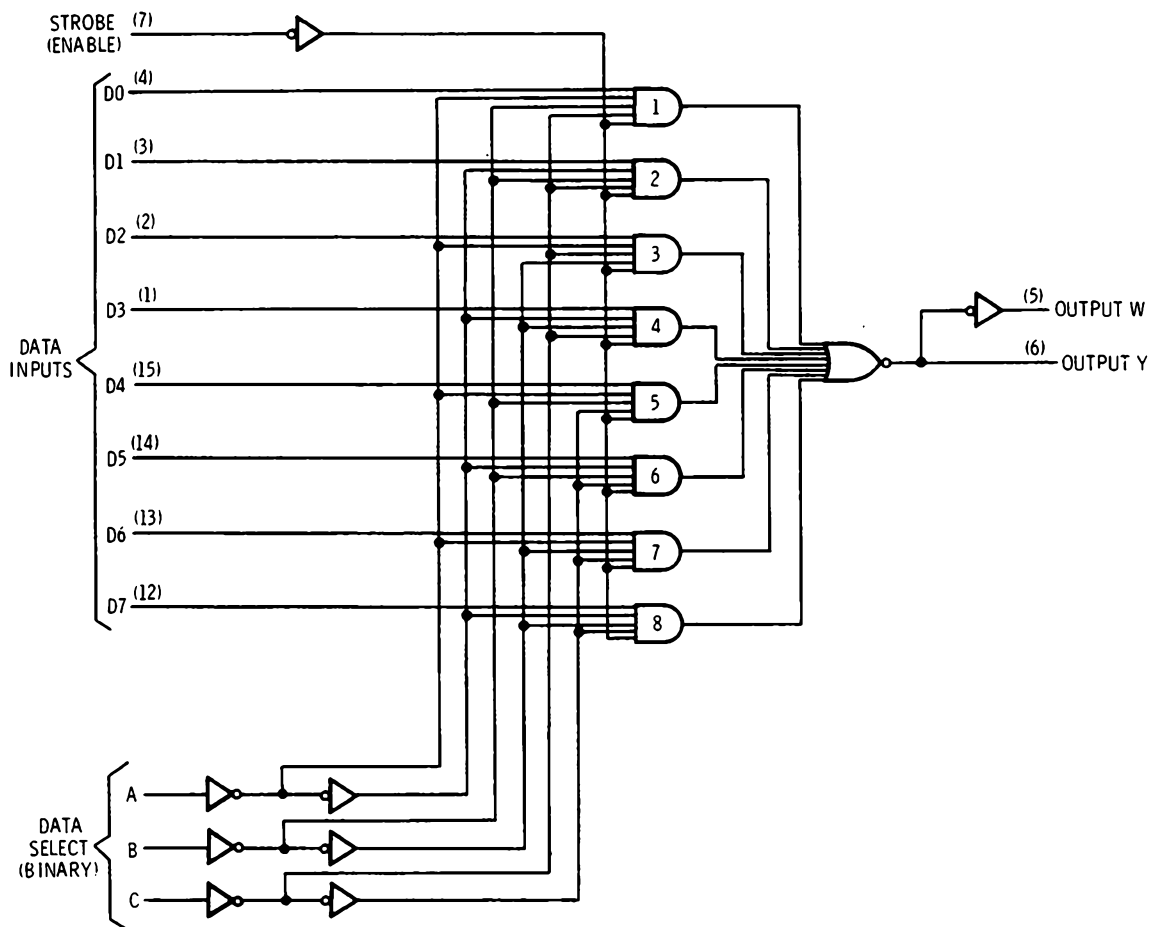
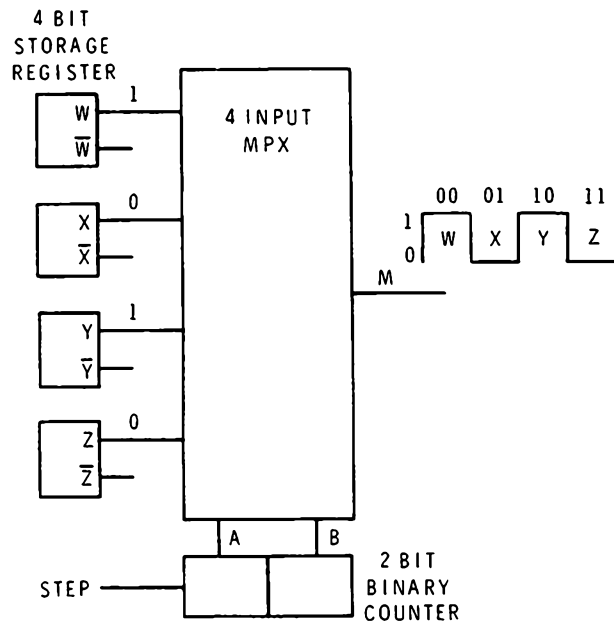


Figure 8-25
8-input multiplexer
TTL IC type 74LS151.

Multiplexer Applications

Besides providing a convenient means of selecting one of several inputs to be connected to its single output, a multiplexer has several special applications which make it even more useful. Besides its data selector application, multiplexers are also used to provide parallel to serial data conversion, serial pattern generation, and the simplified implementation of Boolean functions. Let's consider these important applications here.

Parallel to Serial Conversion. One of the most common applications of a multiplexer is parallel to serial data conversion. A parallel binary word is applied to the inputs of a multiplexer. Then by sequencing through the input enabling codes, the output of the multiplexer becomes a serial representation of the parallel input word. This function is illustrated in Figure 8-26. Here we show a four input multiplexer. (Multiplexer is often abbreviated MPX or MUX). The simple block diagram is often used to represent multiplexers in order to simplify their illustration. A two bit binary input word AB from a counter is used to select the desired input. Input word WXYZ is stored in a 4 bit storage register. The output of each of the flip-flops in the register is connected to one input of the multiplexer. As the two bit counter is incremented, the AB input select code is sequenced through its four states 00 through 11. The output (M) of the multiplexer is equal to the state of the flip-flop connected to the enabled input. This is illustrated by the truth table in Figure 8-26. By sequencing through the four input states at a fixed rate, the parallel input word is converted to a serial output word. When the AB inputs are 00, the state of the W flip-flop appears at the multiplexer output. When the AB input state is 01, the state of the X flip-flop appears at the multiplexer output. Similarly, input select states 10 and 11 cause the states of flip-flops Y and Z respectively to appear at the multiplexer output. Depending upon how the inputs are connected to the register, the multiplexer can cause either the LSB or the MSB to occur first.



INPUTS		OUTPUTS
A	B	M
0	0	W
0	1	X
1	0	Y
1	1	Z

Figure 8-26
4-input multiplexer used
as a parallel-to-serial converter.

Serial Binary Word Generator. Another application of the multiplexer in digital circuits is the generation of a serial binary word. This application is virtually identical to the parallel to serial conversion technique just discussed. The primary difference is that for binary word generation, the serial word generated at the output of the multiplexer is generally a fixed value rather than one that can change as in the case of the parallel to serial converter. There are some occasions that require the generation of a single fixed serial word for some special function.

Figure 8-27 shows an eight input multiplexer used to generate a fixed serial binary output word. Notice that the eight inputs are connected to either +5 volts (binary 1) or ground (binary 0). The three bit input word ABC is used to select which of the inputs is routed to the output. By sequencing through the three bit input words from 000 through 111 with a binary counter, the binary states applied to inputs 0 through 7 are sequentially connected to the output. The binary word 10011010 is generated at the output. Each time the three bit input word is sequenced through the 000 to 111 state, this serial output word will be generated. Again, depending upon the application, the connections to the multiplexer input can be made such that either the MSB or LSB occurs at the output first. In this case the MSB appears at the output first. The truth table in Figure 8-27 completely defines the function of this circuit.

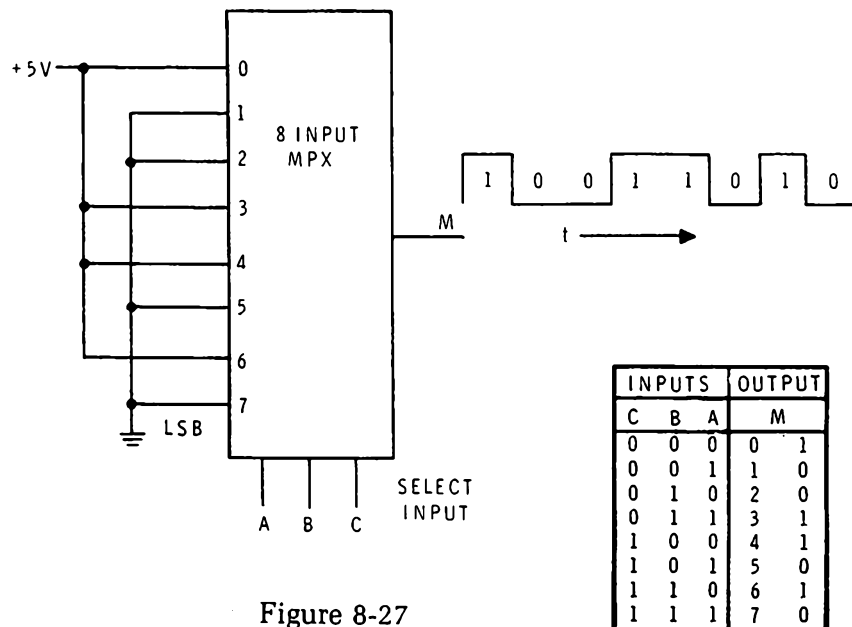


Figure 8-27
8-input multiplexer
used to generate the serial
binary word 10011010.

Boolean Function Generation. Multiplexers can greatly simplify the implementation of Boolean functions in the sum-of-products form. A close look at the multiplexer circuit in Figure 8-25 shows that it inherently implements the sum-of-products for all input combinations. The products $\overline{A}\overline{B}\overline{C}$ through ABC are developed by gates 1 through 8. By connecting a binary 1 or binary 0 to the appropriate data inputs, the products desired in the output can be selected.

For example, suppose that you wish to implement the Boolean function indicated below.

$$M = A\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + ABC$$

By studying the logic diagram for the multiplexer in Figure 8-25, you can determine which gates generate each Boolean product. These are indicated in Table A for your convenience. Note that gate 1 generates the product $\overline{A}\overline{B}\overline{C}$. When a binary 1 is applied to the D_0 input, a binary 1 will appear at the output if the data select inputs are 000. By applying a binary 0 to D_0 , the 0000 input state will be ignored and a binary 0 will appear at the output W .

Table A		
Data Input	Gate	Output
D_0	1	$\overline{A}\overline{B}\overline{C}$
D_1	2	$A\overline{B}\overline{C}$
D_2	3	$\overline{A}B\overline{C}$
D_3	4	$AB\overline{C}$
D_4	5	$\overline{A}\overline{B}C$
D_5	6	$A\overline{B}C$
D_6	7	$\overline{A}BC$
D_7	8	ABC

Therefore, you can see that the desired Boolean products can be selected by applying a binary 1 to the appropriate input associated with the gate generating that product. Those products you wish to delete from the output, you apply a binary 0 to the gate generating that product.

To generate the expression indicated earlier, binary 1 states are applied to the D1, D4, D6, and D7 inputs. These are gates 2, 5, 7, and 8 in Figure 8-25. The complete Boolean function generator is shown in Figure 8-28.

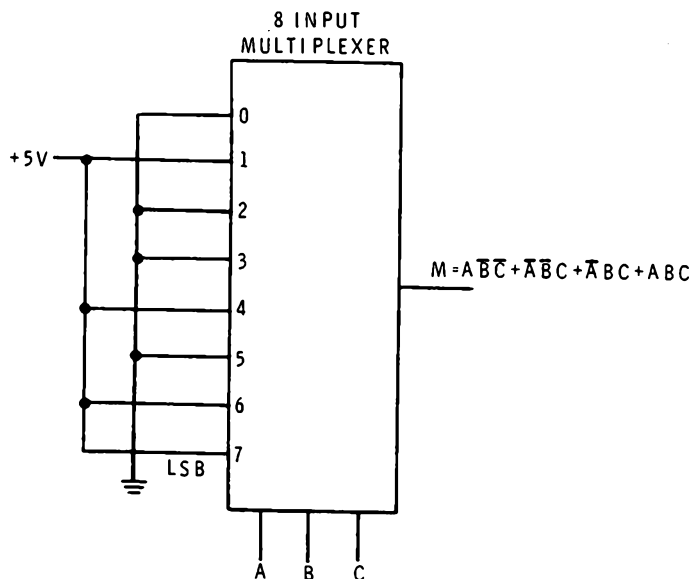


Figure 8-28
Multiplexer used as a Boolean
sum-of-products function generator.

Other more complex Boolean functions can also be implemented with multiplexers by connecting other input variables to the multiplexer inputs instead of a fixed binary 1 or binary 0 level. Four variable sum-of-products from inputs A, B, C, and D for example, can be implemented with an eight input multiplexer by connecting the D and \bar{D} input states to selected multiplexer inputs to implement the desired function.

By using standard MSI multiplexer packages, the implementation of Boolean functions are greatly simplified. With this technique it is not necessary to interconnect multiple SSI logic gate packages to implement the desired function. This greatly reduces the number of integrated circuits used, the power consumption, size, and the need for interconnection.

Self Test Review

11. Which of the following definitions best describes a digital multiplexer?
 - a. a circuit which can route a single input to one of several outputs.
 - b. a circuit that recognizes a specific input code.
 - c. a circuit that connects one of several inputs to any of several outputs.
 - d. a circuit that can connect one of several inputs to a single output.
12. What binary input code must be applied to the data select inputs (ABC) of the 8 input multiplexer in Figure 8-25 (A = LSB) to permit data input D3 to be connected to the output?
 - a. 001
 - b. 101
 - c. 011
 - d. 110
13. What serial binary word is generated by the multiplexer circuit shown in Figure 8-28 if inputs A, B, and C are incremented from 000 through 111?
14. What Boolean function will be generated by the multiplexer circuit shown in Figure 8-27?
15. Another name for a multiplexer is _____.

Answers

11. d. a circuit that can connect one of several inputs to a single output.
12. d. $ABC = 110$ or $CBA = 011 = 3$
13. 01001011 (D0 through D7). As the input select states ABC are stepped from 000 through 111, input D0 through D7 will be enabled in sequence producing a serial binary word that is a function of the input states.
14. $M = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC$
 Binary 1 inputs are applied to data inputs D0, D3, D4, and D6. These are connected to gates 1, 4, 5, and 7 respectively which according to Table A generates the products $\overline{A}\overline{B}\overline{C}$, $A\overline{B}\overline{C}$, $\overline{A}\overline{B}C$, $\overline{A}BC$.
15. data selector

DEMULTIPLEXERS

A demultiplexer is a logic circuit that is basically the reverse of a multiplexer. Where the multiplexer has multiple inputs and a single output, the demultiplexer has a single input and multiple outputs. The input can be connected to any one of the multiple outputs. The demultiplexer is also known as a data distributor or data router.

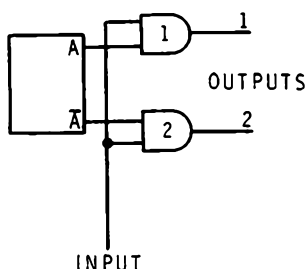


Figure 8-29
2-output demultiplexer.

A simple two output demultiplexer circuit is shown in Figure 8-29. The single input is applied to both AND gates 1 and 2. The A flip-flop selects which gate is enabled. When the A flip-flop is set, gate 1 will be enabled and gate 2 will be inhibited. The input therefore will pass through gate 1 to output number 1. Resetting the flip-flop enables gate 2 and the input is passed to output number 2.

A four output data distributor is shown in Figure 8-30. Here the single input is applied to four gates simultaneously. As in the multiplexer, additional inputs on the select gates are used for decoding. A two bit word AB from a counter is used to select which gate is enabled. If the two input binary word AB is 11, gate 4 will be enabled and the input will pass through gate 4. The other three gates will be inhibited at this time.

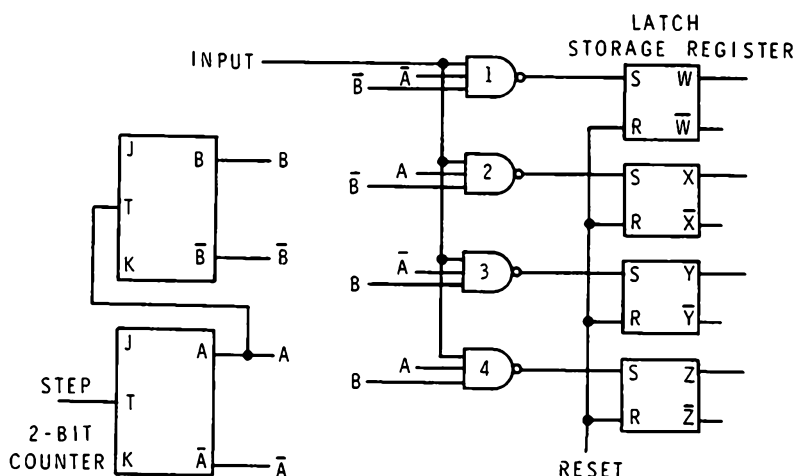


Figure 8-30
A 4-output demultiplexer used
as a serial-to-parallel converter.

The data distributor shown in Figure 8-30 is being used as a serial to parallel converter. This is one typical application of a demultiplexer circuit. A four bit serial word is applied to the input. As the input bits occur, the two bit counter is incremented. This causes the gates in the distributor to be enabled one at a time, sequentially from top to bottom. The step input to the two bit counter is in synchronism with the occurrence of the bits in the serial word.

The latch storage register with flip-flops WXYZ is initially reset prior to the application of the serial input. The flip-flops in the storage register are connected to the output of the data distributor and are sequentially set or left reset as the serial word occurs. Once each of the four gates has been enabled in sequence, the register contains the serial input word. Its outputs can then be observed simultaneously. The serial input word has been converted to a parallel output word.

Figure 8-31 shows the waveforms of the circuit in Figure 8-30. The input is the serial number 1101. The waveforms show the outputs of gates 1 through 4 and the flip-flop outputs WXYZ. The first bit of the serial input is a binary 1. It occurs during the $\bar{A}\bar{B}$ input sequence. During this time, gate 1 is enabled and, since the input is a binary 1, its output will go low. This will set the W latch, causing the W output to go high. The $A\bar{B}$ input selection sequence is next. Note that this is synchronized with the next input, which is also a binary 1. This input state causes gate 2 to be enabled. Since the input signal is a binary 1 at this time, the output of gate 2 will go low, thereby setting the X flip-flop. The X output goes high as indicated. During the next input selection sequence $\bar{A}B$, the serial input word is 0. Gate 3 is enabled. The input is binary 0 at this time so the output of gate 3 remains high. This has no effect on the Y flip-flop so it remains reset. The AB input selection sequence is next. It occurs in synchronism with the next serial bit which is a binary 1. Gate 4 is enabled and, with the binary 1 input, its output is low. This sets the Z flip-flop, causing its output to go high. Looking at the states of the flip-flops after the fourth serial input bit has occurred, you can see that the parallel output 1101 is available. Note that all of the reset inputs to the latch flip-flops in the storage register are connected together to form a common reset line. Prior to the application of the serial input, a low signal is applied to the reset input to clear the register to the 0000 state.

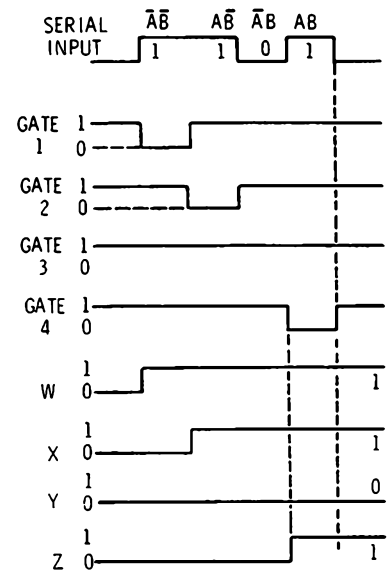


Figure 8-31
Waveforms of a
serial-to-parallel conversion
with a demultiplexer.

A close look at the data distributor circuit in Figure 8-30 shows that it is essentially a decoder where the decode gates all have a common input. Because of this particular configuration, a standard MSI decoder circuit can often be used as a data distributor. Figure 8-32 shows how a 7442 BCD to decimal decoder can be used as an 8 output data distributor. When this circuit is used as a data distributor, inputs A, B, and C are used to select the desired output. These three inputs will enable one of the gates 1 through 8. The data input is applied to the D input of the circuit. Note that data input is inverted by inverter 17 and then applied to gates 1 through 8. The data input will appear at the output of the gate selected by the three bit input word ABC. For example, if the input state is 000, gate 1 will be enabled. The data applied to the D input will appear at the output of gate number 1. In this application gates 9 and 10 of the decoder are not used.

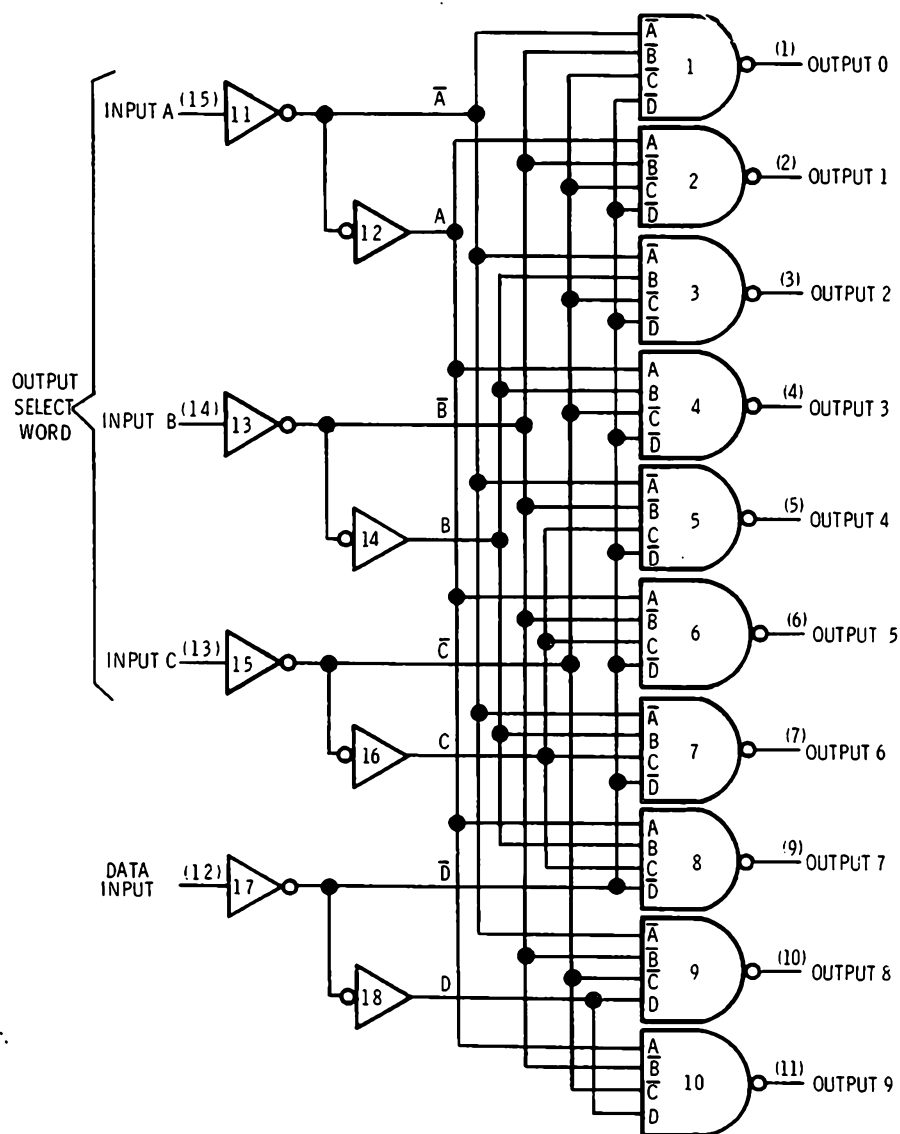


Figure 8-32
A 7442 decoder used as
an 8-output data distributor.

Self Test Review

16. Another name for a demultiplexer is _____.
17. A typical application for a demultiplexer is _____.
18. In Figure 8-30, if A is set and B is reset and the input is binary 1 gate _____ will be enabled and latch _____ will be binary _____.
19. In Figure 8-32 what input code (CBA) must be applied to connect the input to the output of gate 6?
 - a. 010
 - b. 011
 - c. 101
 - d. 110

Answers

16. data distributor or data router
17. serial to parallel conversion
18. 2, X, 1
19. c. 101

EXCLUSIVE OR

One of the most widely used of all combinational logic circuits is the exclusive OR. It occurs so frequently in logic circuits that it is often considered to be one of the basic logic functions such as AND, OR, and NOT. The exclusive OR is a two input combinational logic circuit that produces a binary 1 output when one, but not both, of its inputs is binary 1.

The standard OR logic circuit is generally referred to as an inclusive OR. The OR circuit produces a binary 1 output if any one or more of its inputs are binary 1. The exclusive OR produces a binary 1 output only if the two inputs are complementary. The table below compares the output for the standard inclusive OR and exclusive OR circuits. The inputs are A and B, the output is C.

Inclusive OR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Exclusive OR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

The exclusive OR logic function can be written as a Boolean expression. By using the technique you learned earlier, you can write the logic equation from the truth table. By observing the input conditions that produce binary 1 outputs, you can write the sum-of-products output. The exclusive OR function is indicated below.

$$C = \bar{A} B + A \bar{B}$$

A special symbol is used to designate the exclusive OR function in Boolean expressions. Like the plus sign represents OR and the dot represents the AND function, the symbol \oplus represents the exclusive OR function. The exclusive OR of inputs A and B is expressed as indicated below.

$$C = A \oplus B = \bar{A} B + A \bar{B}$$

The exclusive OR function can be simply implemented with standard AND and OR gates as shown in Figure 8-33. The expression X-OR is often used as a short hand method for indicating the exclusive OR function.

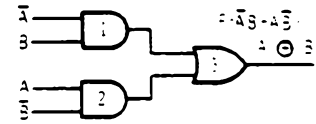


Figure 8-33
Basic exclusive OR logic circuit.

Figure 8-34 shows several ways of implementing the exclusive OR function with NAND and NOR gates. The exclusive OR function implemented with NAND gates is illustrated in Figure 8-34A. The NOR implementation of the X-OR function is shown in Figure 8-34B.

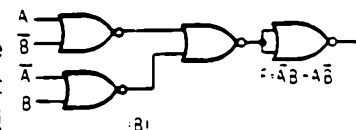
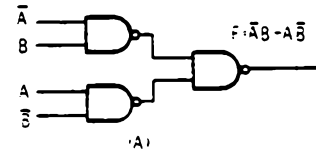


Figure 8-34
Implementation of the
X-OR function with NAND
gates (A) and NOR gates (B).

Figure 8-35 shows how the exclusive OR function can be performed using the wired AND connection. Here the open collector outputs of TTL gates can be connected together to produce the AND function required by the X-OR operation.

The exclusive OR circuits in Figures 8-33, 8-34, and 8-35 all assume that both the normal and complement versions of the A and B input signals are available. If they are not, then input inverters can be used to produce them. In some circuits this means extra components and a greater number of interconnections. The exclusive OR circuit in Figure 8-36 avoids this problem. Only the A and B input signals are required in order to generate the X-OR function at the output. This circuit can be readily constructed for example from a standard quad two input NAND gate such as the TTL 74LS00.

In order to avoid the necessity of drawing the exact logic diagram for each exclusive OR circuit used, the simplified symbol shown in Figure 8-37 is used. This symbol can be used to represent any of the exclusive OR circuits that we have described so far.

With modern integrated circuits, it is generally not necessary to actually construct exclusive OR circuits from individual logic gates. Instead exclusive OR circuits are available in MSI form. For example, the 74LS86 TTL integrated circuit contains four completely independent X-OR circuits.

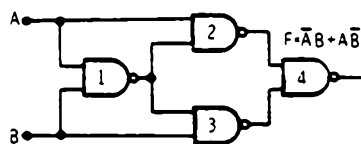


Figure 8-36
X-OR circuit not
requiring complement inputs.

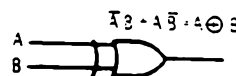


Figure 8-37
Standard symbol for
an exclusive OR circuit.

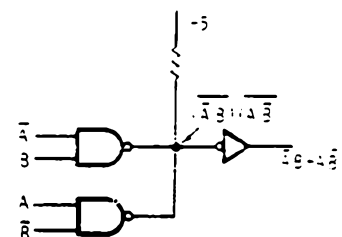


Figure 8-35
The X-OR function complemented
with the wired-AND connection.

Exclusive NOR

An often used version of the exclusive OR is the exclusive NOR (X-NOR) circuit. The truth table for this circuit is given below.

A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

Note that the output of the equivalent circuit is a binary 1 when inputs A and B are equal. If both inputs are 0 or both inputs are 1, the output will be a binary 1. As a result, the exclusive NOR is sometimes referred to as an equivalence circuit or a comparator. Comparing this to the exclusive OR function you can see from the truth table that the output of the X-NOR is the complement of the X-OR.

The Boolean equation of the X-NOR circuit can be written from the truth table. It is indicated below.

$$C = \overline{A} \overline{B} + AB$$

Since the form of this equation is similar to that for the exclusive OR, (sum-of-products) the equivalence function can be implemented by using any one of the exclusive OR circuits given previously by simply rearranging the inputs. Alternately, all of the previously given exclusive OR circuits can also be used to perform the equivalence operation by leaving the inputs as designated and complementing the output. Figure 8-38 shows several methods of implementing the exclusive NOR function. The simplified symbol shown in Figure 8-39 is frequently used to indicate the X-NOR operation.

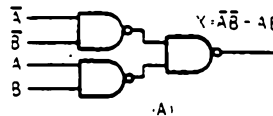


Figure 8-38
Methods of implementing of
exclusive NOR or equivalence
function with NAND gates (A)
and NOR gates (B).

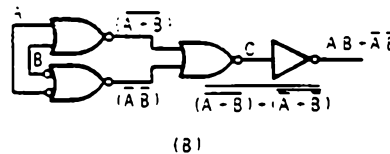


Figure 8-39
Symbol for the
exclusive NOR function.

Applications of the Exclusive OR

As indicated earlier there are many applications for the exclusive OR logic circuit. There are many special combinational circuits that take advantage of the special characteristics of the exclusive OR. Let's take a look at some of the most widely used applications of the exclusive OR and the exclusive NOR circuits.

Binary Adder. A binary adder is a circuit that adds two binary numbers. The output of the adder is the sum of the two input numbers. A binary adder is the basic computational circuit used in digital computers, electronic calculators, microprocessors, and other digital equipment employing mathematical operations.

The basic rules for binary addition are very simple. These are indicated below.

$$\begin{array}{cccc}
 0 & 0 & 1 & 1 \\
 +0 & +1 & +0 & +1 \\
 \hline
 0 & 1 & 1 & 10
 \end{array}$$

↑ carry

These rules indicate how two single bit numbers are added. Naturally, these rules can be extended to multibit numbers. Several examples of the addition of multibit numbers are shown below.

$ \begin{array}{r} 1010 \\ + 3 0011 \\ \hline 13 1101 \end{array} $	$ \begin{array}{r} 0111 \\ + 11 1011 \\ \hline 18 10010 \end{array} $	$ \begin{array}{r} 1100 \\ + 10 1010 \\ \hline 22 10110 \end{array} $
---	---	---

A close look at the rules for binary addition indicated above show that if put in truth table form they would be identical to the logical function of an exclusive OR circuit. The exclusive OR inputs A and B are the two single bits to be added while exclusive OR output C is the single bit sum. As you can see, the exclusive OR is a binary adder. The only function not taken care of by the exclusive OR circuit is the carry function. When you're adding two binary 1 bits, a binary 1 carry will be generated. This carry operation can be simply implemented with an AND gate that will produce a binary 1 output only when both inputs are binary 1. Combining the AND gate and the

exclusive OR we can develop a basic single bit binary adder circuit as shown in Figure 8-40. This circuit is generally referred to as a half adder.

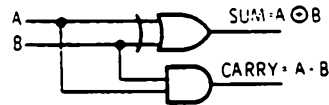


Figure 8-40
Half adder circuit.

To add multibit numbers, we must provide an adder circuit for each of the two corresponding bits to be added. However, the half adder circuit does not provide for a carry input from a lower order bit position. Therefore, an adder circuit must be developed that will add together the two input bits then add to that sum the carry from the next least significant bit position. Such a circuit combines two half adder circuits to form a full adder. This circuit is shown in Figure 8-41. The half adder made up of exclusive OR gate 4 and AND gate number 1 performs the addition of the two input bits A and B. The output of exclusive OR gate 4 is the sum of these two bits. To this sum is added the carry input (C_i) from the adjacent lower order bit position. The sum of bits A and B is added to the carry input in the half adder circuit made up of exclusive OR gate 5 and AND gate 2. The output of exclusive OR gate 5 is the correct sum. Note that because two half adders are used there will be two carry outputs. Since a carry can be generated from either the addition of the two inputs A and B or the addition of their sum and the carry, the two carry outputs are ORed together in gate 3 to produce a correct carry output (C_o) that will feed the next most significant bit adder in a multibit adder.

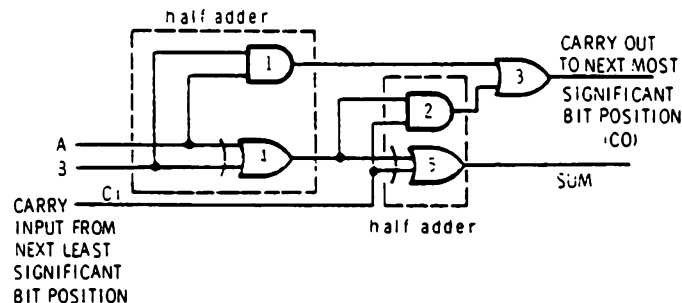


Figure 8-41
Full adder circuit.

Figure 8-42 shows a block diagram of an adder circuit used to produce the sum of two four bit binary numbers. The inputs are two four bit binary numbers A and B. Input number A is made up of bits A1, A2, A3, and A4. Input number B consists of bits B1, B2, B3, and B4. Each of the corresponding bits of the two numbers is added or summed in an adder circuit. Note that the least significant bits A1 and B1 are added in a half adder. Since there is no lesser significant bit, no carry input is required and a half adder circuit will suffice. All other bit positions require a full adder circuit to accommodate the carry input from the next lower order bit position. The output is a four bit parallel sum of the two input numbers with bits S1, S2, S3, and S4. The carry output of the most significant bit full adder also represents the fifth or most significant output bit in those situations where the four bit input numbers produce a five bit sum.

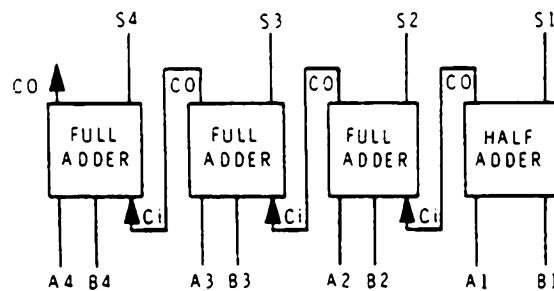


Figure 8-42
4-bit parallel adder.

While the adder circuits described here can be constructed of exclusive OR gates and other logic elements, it is generally unnecessary since single bit adders and four bit adders like those discussed here are available as complete MSI integrated circuits in a single easy to use package.

Parity Generator/Checker. A parity generator is a combinational logic circuit that generates a single output that indicates the presence or absence of a bit error in a binary word. In digital applications requiring the storage of binary data in an electronic memory or in the transmission of binary information from one location to another, there is the likelihood of an error being made. Because of electrical noise or circuit failure, a binary 1 bit may be stored or transmitted as a binary 0. A binary 0 bit could be stored, transmitted or received as a binary 1. In most electronic equipment it is desirable to know when such errors occur. A parity generator circuit performs this function.

The parity generator circuit looks at the binary word to be stored or transmitted and generates a single output known as a parity bit. This parity bit is then added to the other bits of the word and stored or transmitted with it. When the stored word is retrieved from memory for use or when a transmitted word has been received, a parity check operation is performed. The parity checker generates a parity bit from the received data and compares this bit with the parity bit stored or transmitted with the original information. If the two parity bits are identical, no error exists. A difference in parity bits designates an error.

The method of generating a parity bit is to observe the binary word to be stored or transmitted and determine the number of binary 1's in that word. A parity bit will be generated based on this information such that the total number of binary 1's in the word including the parity bit will be either odd or even. The table in Figure 8-43 shows all 16 possible combinations of four bit binary words. The odd and even parity bits for these words are designated in the adjacent columns. A binary 1 or binary 0 parity bit is added to make the total number of bits in the word, including the parity bit, odd or even.

A	B	C	D	ODD PARITY	EVEN PARITY
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

Figure 8-43
Odd and even parity
bits for a 4-bit word.

The basic circuit element used to generate the parity bit is the exclusive OR circuit. A look at the truth table for an exclusive OR circuit will reveal that it is basically an odd or even detector circuit. If the two inputs are equal or even, the exclusive OR output is a binary 0. But if the inputs are odd or complementary, the output is a binary 1. The exclusive OR gate then can be used to compare two binary bits and indicate whether they are odd or even, equal or unequal. An exclusive OR gate then is used to monitor each two bit group in a binary word. These exclusive OR outputs are then further compared with other exclusive OR circuits until a single output bit indicating odd or even is generated.

Figure 8-44 shows how exclusive OR gates are cascaded or pyramided to produce a parity generator circuit. A 4-bit binary number input to the parity generator circuit is stored in a register made up of flip-flops A, B, C, and D. X-OR gate 1 monitors bits A and B while X-OR gate 2 monitors bits C and D. The outputs of these two X-OR circuits are then monitored by X-OR gate 3. The result is an even parity output bit. Inverter 4 generates the complement or the odd parity output. Using your knowledge of the exclusive OR circuit and the information in Figure 8-43, trace the various binary states from the flip-flop outputs to the output of the parity generator circuit to be sure that you fully understand its operation.

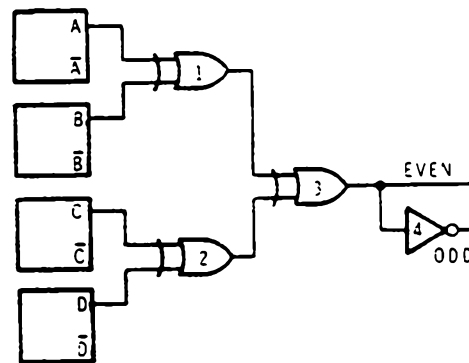


Figure 8-44.
A 4-bit parity generator circuit.

A parity bit for any size binary word can be generated by simply using as many exclusive OR gates as necessary to monitor all input bits. Additional exclusive OR gates are then used to monitor the output states of the exclusive OR gates used in monitoring the input bits. This cascading or pyramiding of exclusive OR gates is continued until a single output bit is generated.

Once the parity bit has been generated, it is generally stored or transmitted along with the input word. When the binary word and its parity bit are read from memory or received at the remote location, it can be tested for bit errors in a parity checker circuit. A parity checker consists of a parity generator circuit identical to those just discussed. This parity generator looks at the stored or received word and again generates a parity bit. This bit is then compared to the parity bit stored or transmitted along with the word. This comparison takes place in another exclusive OR circuit. Figure 8-45 shows a parity checker circuit for a 4-bit binary word with parity bit.

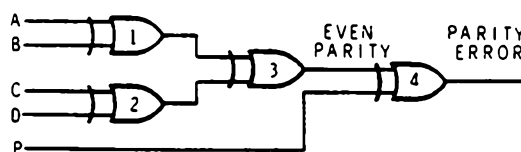


Figure 8-45
A 4-bit parity checker.

Note that the parity generator circuit consists of exclusive OR gates 1, 2, and 3 and is identical to the parity generator circuit in Figure 8-44. Exclusive OR gate 4 compares the output of the parity generator with the received parity bit *P*. In this circuit we are assuming the use of even parity. If the internally generated parity bit is the same as the received parity bit, the output of the exclusive OR circuit will be binary 0 indicating no parity error. However, if the two parity bits are different, the exclusive OR output will be binary 1 indicating a parity error.

The output of a parity checker circuit can then be used in a variety of ways to indicate the occurrence of a parity error. It can be used to turn on an indicator light indicating an error state. It can be used to initiate a series of logic operations that will either accept or reject the data depending on the error state. Or it may be desirable simply to count and record the number of parity errors that occur.

As you probably realize, a parity error detection method does not ensure complete freedom from or knowledge about all possible error conditions. The parity technique assumes that an error will occur in only one bit position of a word. If parity errors occur in two bit positions, it is possible for the word to be transmitted incorrectly while no parity error will be indicated. This situation rarely occurs since in most electronic storage and transmission systems the reliability is sufficient to eliminate the possibility of multibit errors. However, errors in a single bit position are common. The parity detection and checking process is a very reliable and useful indication of errors.

Even more sophisticated combinational logic circuits have been developed to detect when more than one bit error is produced. In systems requiring ultra high reliability and performance, such sophisticated circuits can be used to detect and even correct any bit error that occurs. In some high speed computers, multiple bit errors are automatically detected and corrected before the information is processed.

While parity generator and checker circuits can be constructed with individual exclusive OR gates, they are also available in MSI form. Figure 8-46 shows a typical commercial parity generator/checker MSI circuit. This circuit is capable of performing either the generation or checking function. As a generator, it can monitor up to nine input bits. The ninth bit is applied to either the odd or even input as required by the application. Both odd and even parity bit outputs are provided. In the checking function, this circuit will monitor an 8 bit word and generate an appropriate parity bit which is then compared with a received parity bit applied to either the odd or even input. The error indication appears at the odd or even output depending upon whether the odd or even parity convention is used.

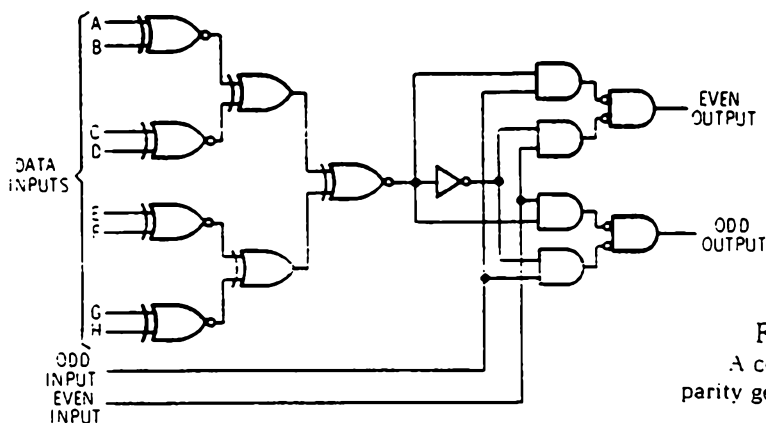


Figure 8-46
A commercial MSI
parity generator/checker IC.

Binary Comparators. A binary comparator is a combinational logic circuit that looks at two parallel binary input words and generates a binary 1 output signal if the two numbers are equal. If the numbers or words are not the same, the output will be a binary 0.

As you saw earlier, the exclusive NOR circuit is essentially a single bit binary comparator. When the two inputs are alike, the output is binary 1. When the input bits are different, the output is binary 0. By using an exclusive NOR circuit for each pair of bits in the two numbers to be compared, a complete binary comparator circuit can be constructed.

Figure 8-47 shows a four bit binary comparator circuit. Word 1 with bits A1, A2, A3, and A4 are stored in the A register. The word to be compared is stored in the B register with bits B1, B2, B3, and B4. Each pair of bits is applied to an exclusive NOR circuit. The outputs of the exclusive NOR's are fed to a four input AND gate. When the two binary words are alike, the outputs of the exclusive NOR's will be binary 1. With all binary 1 inputs to the AND gate, the output will be binary 1 indicating the equality of the two words. If any one or more bits of the input words are different, the output of the related exclusive NOR circuit will be binary 0. Naturally, this will inhibit the AND gate and produce a binary 0 output which indicates inequality. Additional exclusive NOR gates and AND gates inputs can be added as required to compare any size binary number.

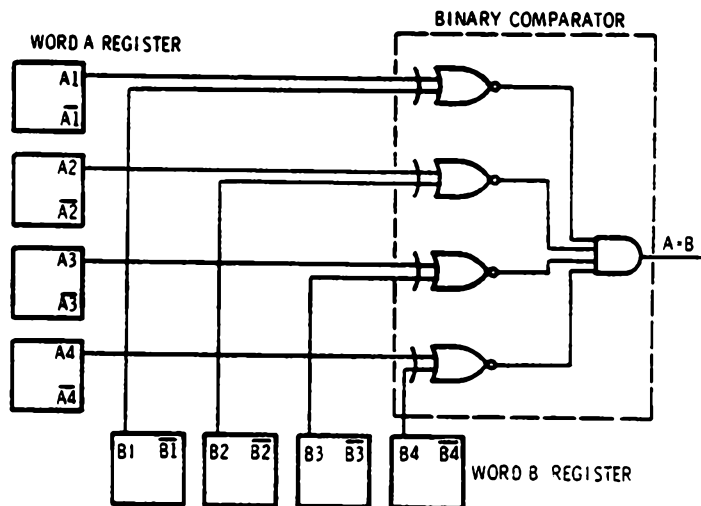


Figure 8-47
4-bit binary comparator.

Commercially available MSI binary comparators are available thus eliminating the need to assemble such circuits from individual gates. Typically, these comparators are designed for comparing two four bit binary words. In addition to providing an output that indicates the equality of the two words, most MSI comparators also generate two additional output signals, one indicating when one word is greater than the other and another indicating when one word is less than another. Figure 8-48 shows a block diagram of such a comparator. If input word A has a binary value that is numerically larger than word B, the A greater than B output ($A > B$) will be binary 1. the $A < B$ output will be binary 1 if A is less than B.

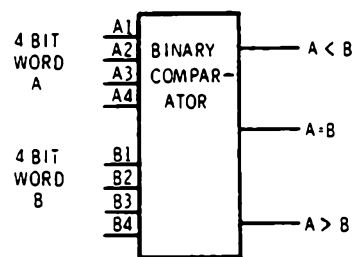


Figure 8-48
Typical 4-bit commercial
MSI binary comparator circuit.

Self Test Review

20. When the inputs to an exclusive OR circuit are alike, the output is binary _____.
21. Prove that the circuit in Figure 8-36 does perform the exclusive OR function. Write the output equation of the circuit using the NAND relationship for each gate. Then using Boolean algebra and DeMorgan's theorem, reduce this expression to the exclusive OR formula.
22. Using Boolean algebra and DeMorgan's theorem, show that the complement of the exclusive OR function is the equivalence function. Or prove that:

$$\overline{A B + A \overline{B}} = \overline{A} \overline{B} + A B$$

23. An exclusive NOR gate is also a(n)
- adder
 - comparator
 - subtractor
 - decoder
24. The expression $\overline{A \oplus B}$ indicates the
- exclusive OR
 - inclusive OR
 - exclusive NOR
25. Add the following binary numbers.

a. 011	b. 11111	c. 1010
<u>+101</u>	<u>+10001</u>	<u>+1011</u>

26. Write the odd parity bit code for the Gray code.
27. How many exclusive NOR's are needed to make a comparator for two six bit words?
- 2
 - 3
 - 6
 - 12

Answers

20. binary 0

21. Refer to Figure 8-36.

$$\text{Output of gate 1} = \overline{AB}$$

$$\text{Output of gate 2} = \overline{A \cdot \overline{AB}}$$

$$\text{Output of gate 3} = \overline{B \cdot \overline{AB}}$$

$$\text{Output of gate 4} = C = \overline{(\overline{A \cdot \overline{AB}})(\overline{B \cdot \overline{AB}})}$$

Apply DeMorgan's and Double Negative rules

$$C = (A \cdot \overline{AB})(B \cdot \overline{AB})$$

$$C = (A \cdot \overline{AB}) + (B \cdot \overline{AB}) =$$

Expand by DeMorgan's

$$C = A \cdot (\overline{A} + \overline{B}) + B(\overline{A} + \overline{B})$$

Expand by Law of Distribution

$$C = A\overline{A} + A\overline{B} + \overline{A}B + B\overline{B}$$

Reduce by Law of Complements

$$C = A\overline{B} + \overline{A}B$$

22. $\overline{\overline{A}B} + \overline{A\overline{B}} =$

Expand By DeMorgan's

$$(\overline{\overline{A}B})(\overline{A\overline{B}}) =$$

Expand by DeMorgan's

$$(A + \overline{B})(\overline{A} + B) =$$

Expand by Law of Distribution

$$A\overline{A} + AB + \overline{A}\overline{B} + B\overline{B} =$$

Reduce by Law of Complements

$$AB + \overline{A}\overline{B} = \overline{A}\overline{B} + AB$$

23. (b) comparator

24. (c) exclusive NOR = $A \oplus B$

25.	a.	011	3	b.	1111	31	c.	1010	10
		<u>101</u>	<u>+5</u>		<u>10001</u>	<u>+17</u>		<u>1011</u>	<u>+11</u>
		1000	8		110000	48		10101	21

26.

DECIMAL	GRAY	ODD PARITY
0	0000	1
1	0001	0
2	0011	1
3	0010	0
4	0110	1
5	0111	0
6	0101	1
7	0100	0
8	1100	1
9	1101	0
10	1111	1
11	1110	0
12	1010	1
13	1011	0
14	1001	1
15	1000	0

27. (c) 6

CODE CONVERTERS

A code converter is a combinational logic circuit that converts one type of binary code into another. There are many applications in digital systems where two or more different binary codes are used. There are applications where it is desirable to take advantage of the characteristics offered by the different types of codes. By using code converters, the various circuits can be made compatible. Some of the most commonly used code converters are listed below.

Binary to BCD

BCD to binary

Binary to Gray

Gray to Binary

8421 BCD to XS3

XS3 to 8421 BCD

ASCII to EBCDIC

EBCDIC to ASCII

Keep in mind that a combinational logic circuit can be constructed to convert any code into any other code. In a sense, any multibit input-multibit output combinational logic circuit can be considered a code converter. Any combination of input bits can be considered a binary code. In the same way any combination of output bits can also be considered a unique binary code. Using this broad definition of a code converter then circuits such as encoders and decoders could also be considered special forms of code converters. In general, you will find the term code converter used primarily to reference those special circuits indicated in the list above.

The most commonly used code converters are those for converting between binary and BCD. Block diagrams of such converters are shown in Figure 8-49. In Figure 8-49A, a 6 bit pure binary input number is converted into its two digit BCD equivalent output. Any 6 bit input number 000000 through 111111 (0 through 63 decimal) will be converted by the circuit into its BCD output equivalent. For example, the number 63 in binary is 111111, its BCD output equivalent is 0110 0011.

A BCD to binary converter is shown in Figure 8-49B. A two digit BCD input word is converted into a 7 bit pure binary output equivalent. The input numbers can be anything from 00 through 99 or 0000 0000 through 1001 1001. If the BCD input equivalent of the number 99 is applied, the pure binary output number 1100011 will appear at the output. Both binary to BCD and BCD to binary converters are available as MSI combinational IC's, thereby eliminating the need to implement such circuits with gates.

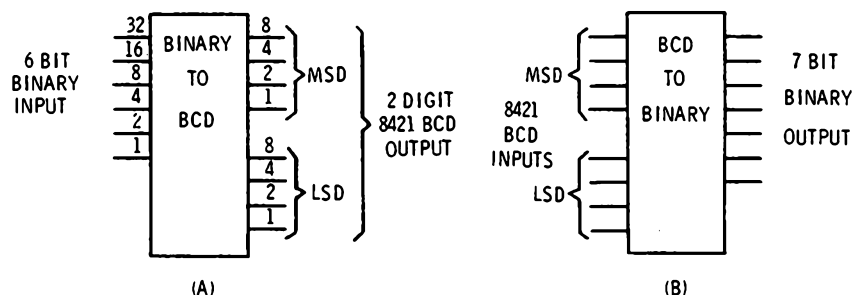


Figure 8-49
Binary to BCD (A) and BCD to binary (B)
code converters.

Other frequently used code converters are binary to Gray and Gray to binary. There are many applications where the Gray or cyclical code must be used in order to minimize errors when changing from one state to another. While the Gray code is good for minimizing errors in generating certain types of data, the Gray code cannot be used in arithmetic operations. To permit arithmetic operations to be performed, Gray to binary code conversion is necessary.

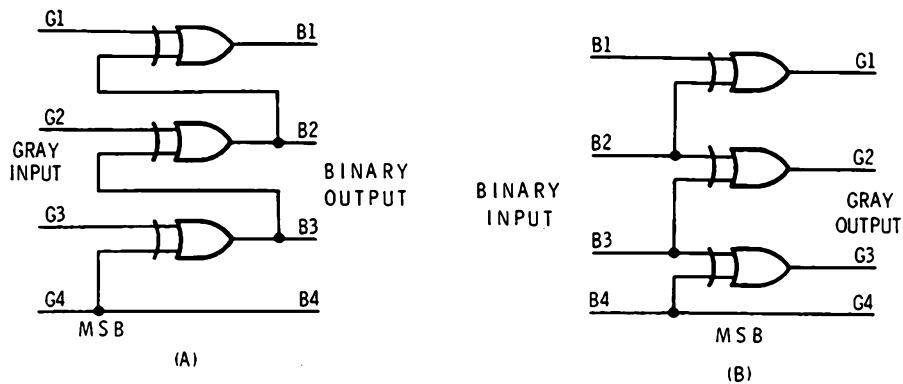


Figure 8-50
Gray to binary (A) and
binary to Gray (B) code converters.

Both Gray to binary and binary to Gray code converter circuits are shown in Figure 8-50. The Gray to binary circuit is shown in Figure 8-50A while the binary to Gray circuit is shown in Figure 8-50B. The most significant bit of both the Gray and binary words will be the same and therefore no code conversion is necessary. Note the use of exclusive OR circuits to perform the code conversion. The Gray and equivalent binary codes are shown in Figure 8-51.

DECIMAL	BINARY				GRAY			
D	B4	B3	B2	B1	G4	G3	G2	G1
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Figure 8-51
Binary and Gray codes.

While code conversion is most often accomplished with combinational logic circuits, many types of code converters use sequential circuits. Various combinations of flip-flops, counters and shift registers can be used to perform code conversion. A simple example is the serial Gray to binary code converter shown in Figure 8-52. Here a serial Gray code is applied to the JK inputs of a JK flip-flop, MSB first. The normal flip-flop output is a serial binary code.

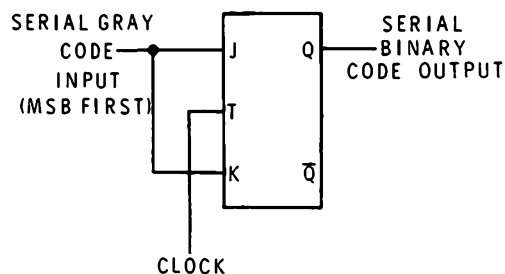


Figure 8-52
Serial Gray to binary
sequential code converter.

Self Test Review

28. The most commonly used code converters use the _____ and _____ codes.
29. Code converters can be either combinational or sequential logic circuits.
 - a. True
 - b. False
30. Code converters can process either serial or parallel data.
 - a. True
 - b. False
31. Most code converters are
 - a. sequential
 - b. combinational

Answers

28. binary, BCD
29. a. True
30. a. True
31. b. combinational

READ ONLY MEMORIES

A read only memory (ROM) is an electronic circuit used to permanently store binary information. Practical read only memories are available for storing in excess of 128,000 bits of data. Normally, the ROM is organized to store equal length multibit words. For example, a typical ROM might be capable of storing 1024 eight bit words of 8192 bits.

The main feature of a read only memory is that the binary information contained in the memory is permanently stored there. The data is written into the memory when it is manufactured, or programmed into the ROM prior to its use. The contents of the memory are not usually changed thereafter. This is in contrast to other types of electronic read/write memories that can both store and read out data. This type of memory is generally referred to as random access memory (RAM). You can think of it as many storage registers.

The general organization of both read/write memories and read only memories is basically the same. Both contain a number of memory locations where data can be stored. In the ROM, the data is stored there permanently and can be read out in any order. In the RAM, data may be written into or read out of any portion of the memory at any time. While the read/write memory is more flexible, it is also more expensive. It is this type of memory that is normally used as the main storage section of a digital computer. Our discussion here will center on the ROM which is a useful circuit in implementing digital systems.

ROM Operation

Figure 8-53 shows a general block diagram of a read only memory. It consists of three major sections: the address decoder, the memory storage elements, and the output circuits.

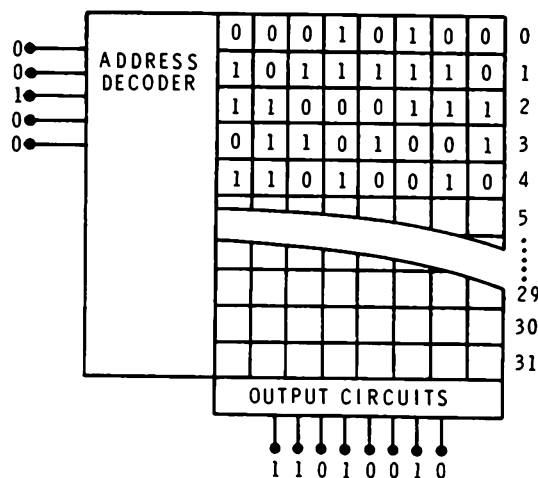


Figure 8-53
General block diagram
of a read only memory.

The address decoder is like any binary decoder in that it accepts a multibit binary input word and decodes all possible input states. Only one of the decoder outputs will be activated. In the address decoder of Figure 8-53, there are five input bits meaning that a total of $2^5 = 32$ different states can be decoded. This five bit input word specifies one of 32 individual memory locations. This input word is generally referred to as the address.

The main body of the memory consists of electronic circuits or components that are used to store the binary data. These storage elements are arranged so that a specific number of multibit binary words may be stored. The organization in Figure 8-53 permits 32 eight bit words to be stored. The memory locations are designated 0 through 31. Applying a five bit address code to the input will cause the contents of the addressed location to appear at the output. Note that if the address input code is 00100, the contents of memory location 4 appear at the output. All other memory locations are ignored at this time. The output circuits buffer the memory contents so that the data can be used in other logic circuits.

ROM Construction

There are many different ways to implement read only memories with electronic components. Any component or circuit capable of storing a binary 1 or binary 0 condition can be used. Magnetic cores and capacitors are examples of elements that have been used to store binary data in a ROM. Most modern read only memories, however, are semiconductor circuits. Both bipolar and MOS types are used. Since ROMs are capable of storing a significant amount of data they are generally classified as large scale integrated (LSI) circuits. Most ROMs, both bipolar and MOS types, are housed in standard dual in-line packages. Because of the wide variety of possible applications, ROMs are considered to be custom circuits. The user specifies the memory contents prior to the manufacture of the device. Special types of ROMs called programmable ROMs or PROMs permit the user to program them electrically himself. In this section, we investigate the most popular types of integrated circuit ROMs in use today.

Diode Matrix ROM. Figure 8-54 shows a read only memory constructed with a one of eight decoder and a diode matrix. The one of eight decoder accepts a three bit address input word and generates all possible decode output combinations. This means that the decoder will recognize the three bit input number applied to it and enable only one of the eight outputs. For example, if the binary input number is 011, the number 3 output line will go low. All other decoder output lines will be high at this time. The decoder is similar in operation to the type 7442 discussed earlier.

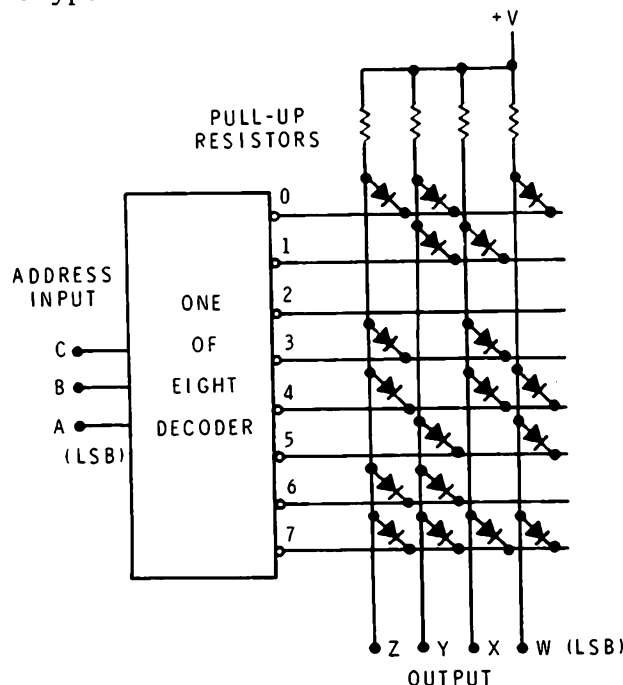


Figure 8-54
A ROM constructed from
a decoder and diode matrix.

When the number 3 decoder output line goes low, it brings the cathode ends of the diodes connected to this line low. The diodes conduct through their associated pull-up resistors. This forces lines X and Z low. Lines W and Y are high at this time because of the pull-up resistors. Since all other decoder outputs are high, the other diodes in the circuit are cut off at this time. Observing the output lines ZYXW then you see the output code 0101. At address location 011 (3), the binary number 0101 is stored.

Consider the effect of applying the address 110 to the decoder input. This will bring decoder output 6 low causing output lines Y and Z to go low. Lines W and X will be high at this time. This means that the output number is 0011. The contents of memory location 110 (6) is the four bit number 0011.

A close look at the ROM in Figure 8-54 should reveal that the data is stored in the memory as either the presence or absence of a diode. In this circuit a diode connection between the decoder output and the output line causes a binary 0 to be read out when that address line is enabled. The absence of a diode causes a binary 1 to be read out. Another way to look at the read only memory is to consider each output line with its associated diodes and pull-up resistors as a diode OR gate. A low on any diode input causes the output to go low.

There are some commercial read only memory ICs designed and constructed exactly like that shown in Figure 8-54. Integrated circuit ROMs are constructed initially so that a diode is connected at each possible memory location. This means that all memory locations are initially programmed with binary 0's. To store data in the memory, an external pulse signal is applied to the output lines in such a manner as to reverse bias certain diodes and cause them to be destroyed. By destroying a diode and causing it to open, a binary 1 state is programmed. Such ROMs can be programmed by the manufacturer or the user. Read only memories that permit the user to store the data that he needs are called programmable read only memories (PROMs).

Bipolar ROM. A typical commercial bipolar read only memory is shown in Figure 8-55. This 256 bit ROM uses TTL circuitry. The memory is organized as 32 eight bit words. The address inputs labeled A through E are used to select one of the 32 words stored in the memory. Note that the circuitry is basically a 1 of 32 decoder.

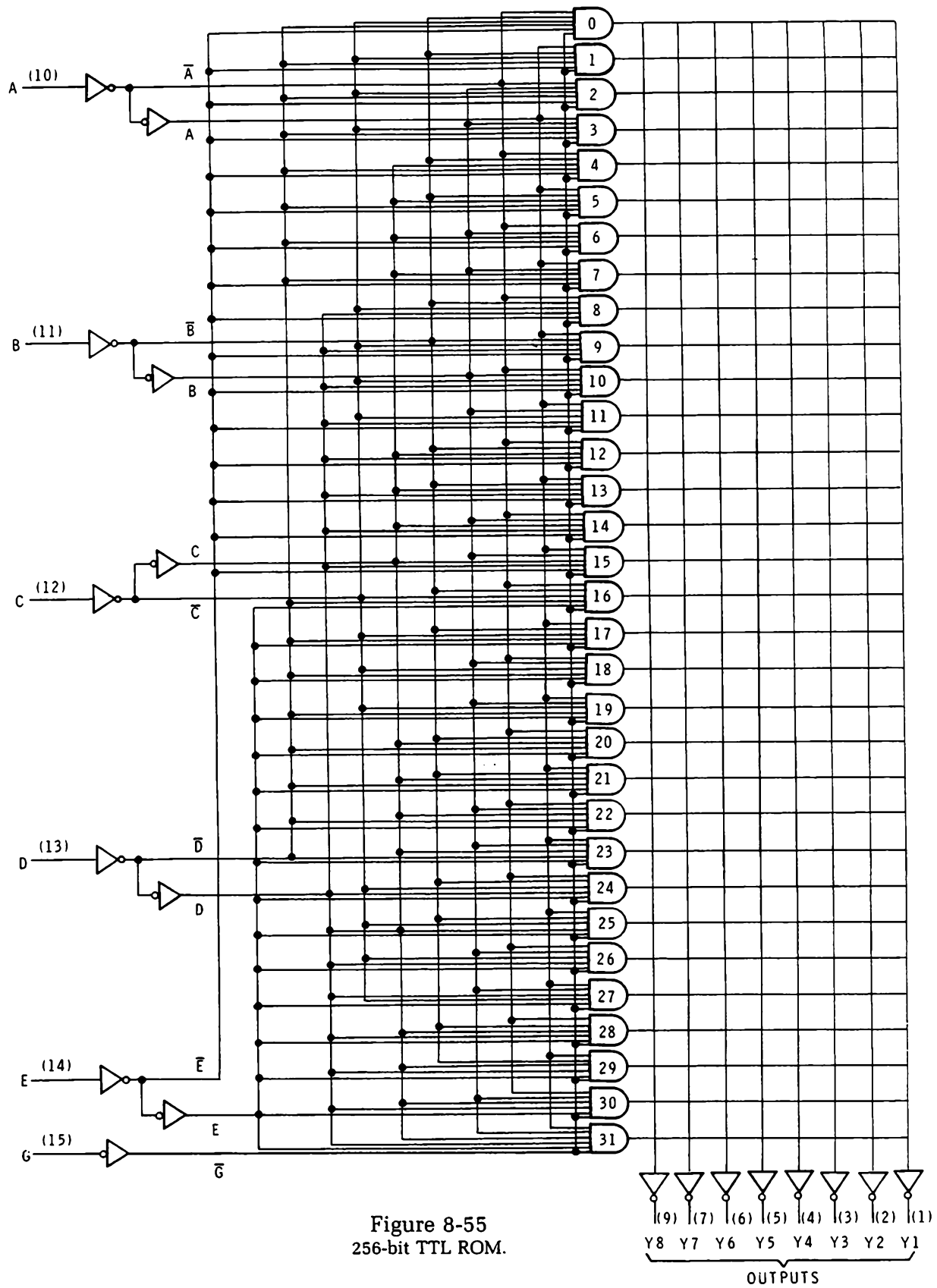


Figure 8-56 shows the detailed circuitry of the ROM. Illustrated here is one of the 32 address decoding gates and the 8 output buffer circuits. The output of each decoding gate is a transistor with eight emitters. These emitters can be interconnected to the eight output buffers. The programming of the memory is done by either connecting or leaving open these emitter connections. If an emitter is connected to an output buffer, the output voltage will go low when that decoding gate is addressed. If the emitter is not connected, a high level voltage is read out of the associated buffer when the gate is addressed.

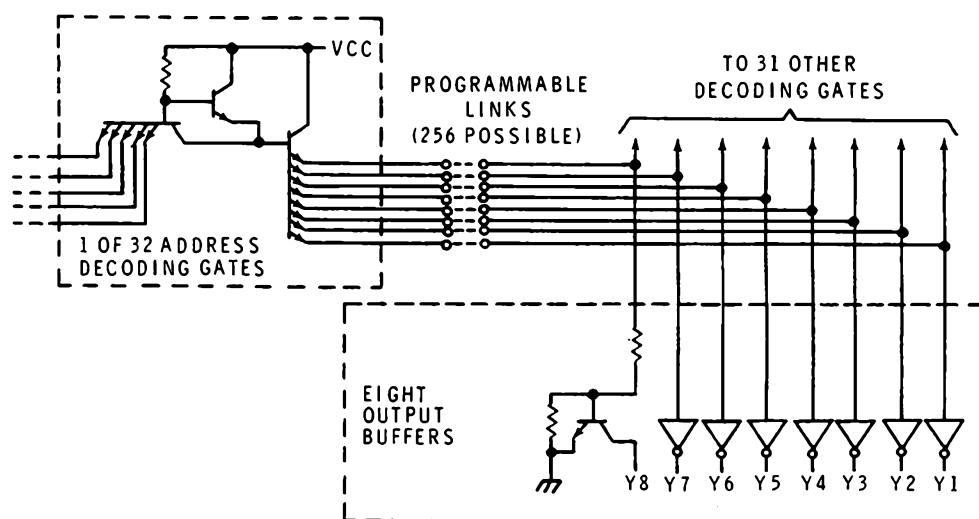


Figure 8-56
TTL ROM circuit details.

The decoding gate output emitters to be used are connected to the respective inputs of the eight output buffers when the integrated circuit is manufactured. The user specifies the memory contents, and the manufacturer produces special masks which will cause the interconnecting metalization of the integrated circuit to be properly arranged to store the desired data. Note that the output buffers have an open collector output. This permits the outputs to be wire-ORed with other similar memories so that the storage capability can be expanded. Three state output circuitry is used on some TTL ROMs. Input line G on the ROM in Figure 8-55 is used to enable or disable the circuitry so that this device can be combined with others similar to it in forming a memory with many more locations. This line is often referred to as a chip select line and is used as an extra address bit input in expanded memories.

Figure 8-57 shows two ways that a standard size ROM can be used to make larger memories. Figure 8-57A shows two ROMs connected to form a memory for 32 sixteen bit words. Each ROM can store 32 eight bit words as indicated by the designation 32 x 8 or 32 by 8. The five address lines are in parallel thus each ROM is enabled at the same time. One half of the 16 bit word is stored in the upper ROM and the other 8 bit segment in the lower ROM. Since the two ROMs are addressed simultaneously, both parts of the word will be read out at the same time. Input line G enables the memory when low.

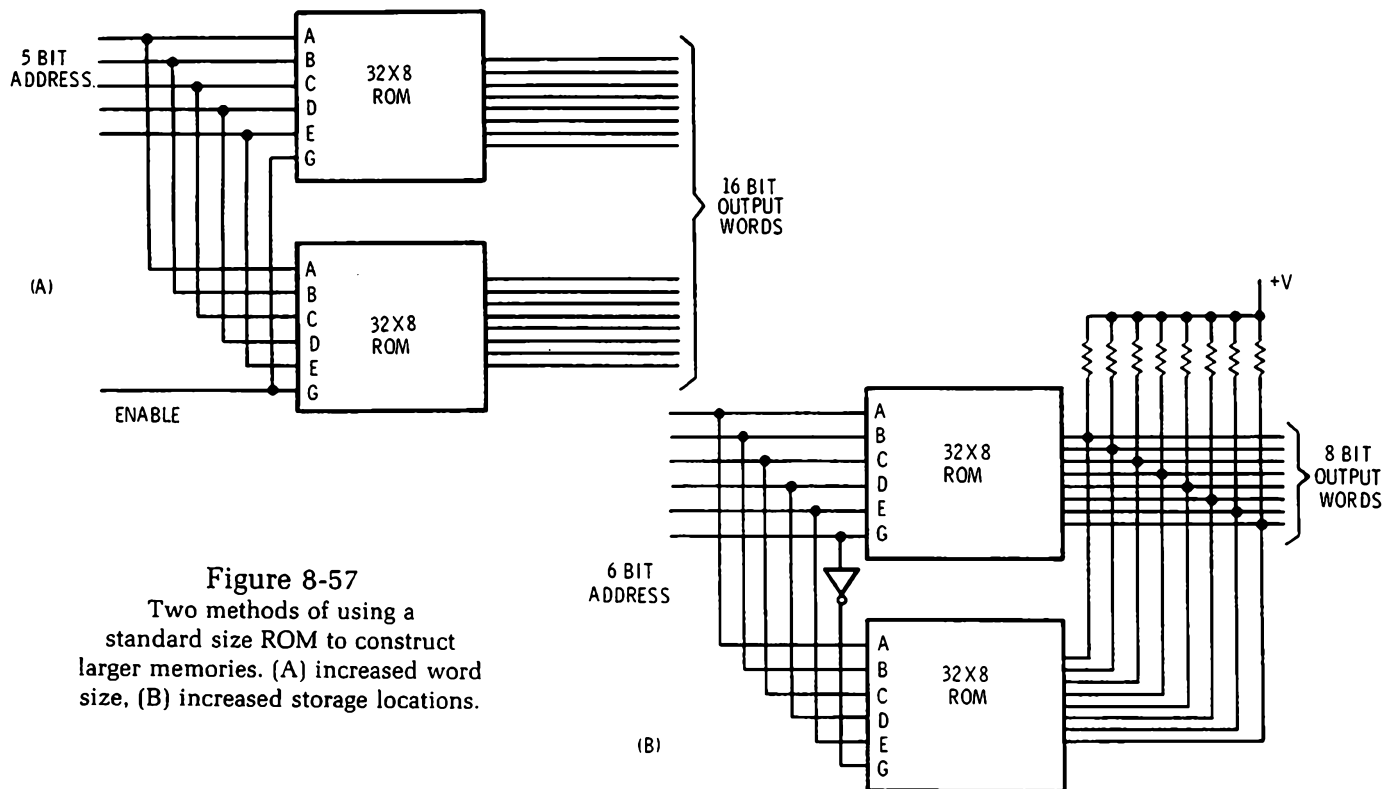


Figure 8-57
Two methods of using a
standard size ROM to construct
larger memories. (A) increased word
size, (B) increased storage locations.

Figure 8-57B shows how the 32 x 8 ROMs can be used to form a memory for storing 64 eight bit words. Thirty-two of the words are in the upper ROM and the other thirty-two are in the lower ROM. The ROM outputs are wire-ORed. The five address lines A through E are in parallel. The chip enable lines (G) are used as a sixth address line. Remember, it takes six bits to address 64 words ($2^6 = 64$). This sixth input line is the MSB of the address. The five lower order bits address both ROMs simultaneously. But only one of the two ROMs will be enabled by input G. The inverter keeps these lines complementary.

If the input address (GEDCBA) 101101 is applied, location 13 (01101) in each ROM will be addressed. However, input G is binary 1. This disables the upper ROM so all of its output lines are high. Input G to the lower ROM is low because of the inverter. Therefore, this ROM is enabled and the word at location 01101 is read out.

MOS ROMs. Many read only memories are implemented with metal oxide semiconductor integrated circuits. MOSFET circuitry lends itself well to the implementation of a read only memory. Because of the small size of most MOSFET circuits, many logic and memory elements can be constructed in a small space. This high density circuitry permits read only memories with a very high bit content to be readily manufactured. Many thousands of bits of data can be stored on a silicon chip approximately 1/10 inch square. Such MOS ROMs are low in cost and consume very little power.

The basic organization and structure of an MOS ROM is essentially the same as any read only memory. An address decoder selects the desired word. The presence or absence of a semiconductor device in a matrix network specifies a binary 1 or binary 0 stored in the addressed location. In MOS ROMs, the basic storage element is an enhancement mode MOSFET. The presence of an MOSFET programs a binary 1. The absence of such a device means a binary 0 has been programmed.

Figure 8-58 shows the basic internal structure of a typical PMOS ROM. P-type material is diffused into the substrate in long strips called bit lines as indicated. These P-type diffusions form the source and drain connections of the MOSFETs. Perpendicular to the P diffusion areas are metal word select lines. These metal areas form the gate elements of the MOSFETs. Figure 8-58 shows several examples of how the MOSFETs are formed. The source (S), gate (G), and drain (D) of each MOSFET is identified. To program the memory, the MOSFETs formed by this structure are either enabled or disabled by appropriate masking operations during manufacturing. As indicated earlier, if the MOSFET is enabled, a binary 1 will be stored in that location. Disabling the MOSFET causes a binary 0 to be stored in the selected location.

In the MOSFET ROM structure, the metal word select lines are connected to the gates of the MOSFETs where binary 1s are stored. These metal word select lines are driven by the outputs of a decoder. The source terminals of the MOSFETs are connected either to ground or to the source supply voltage V_{SS} . The drain connections of the MOSFETs are designated as the bit lines. If one of the metal word select lines goes negative, the MOSFETs associated with that word will conduct and ground (or V_{SS}) will appear on the bit line.

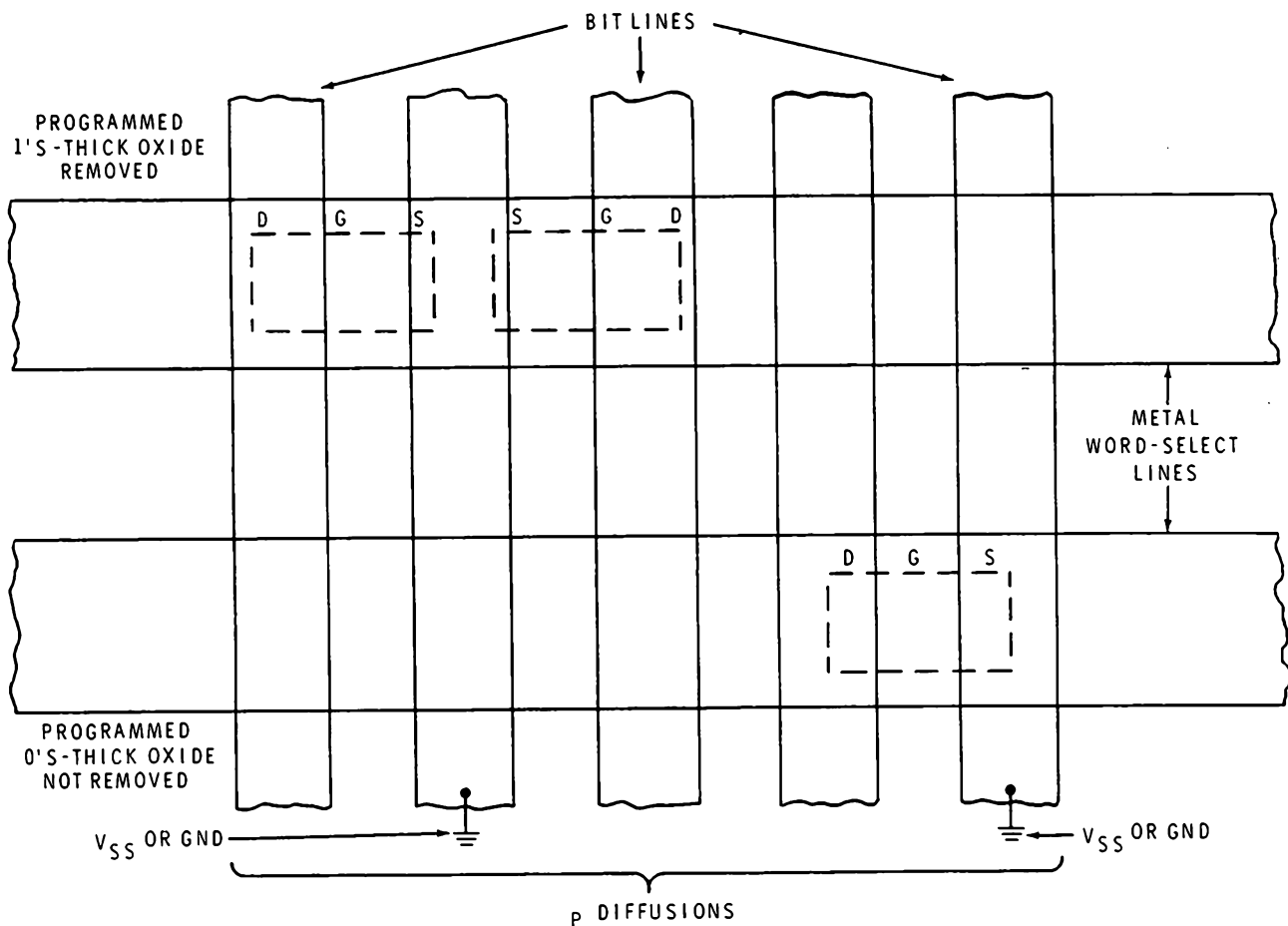


Figure 8-58
Basic structure of a PMOS ROM.

Figure 8-59 shows the MOSFET ROM circuit. Q1 is a MOSFET formed by the process illustrated in Figure 8-58. The presence or absence of this transistor is a function of the masking process carried out during manufacturing. Note that the gate of Q1 is enabled by the output of decoder X. If the word select line is negative, Q1 will conduct and a binary 1 bit will appear on the bit line. However, this binary 1 may or may not reach the output of the ROM depending upon the state of Q2. Q2 and decoder Y are also used in selecting the desired output word.

Most MOS ROMs use an XY matrix decoding method. In Figure 8-59, two 1 of 8 decoders are illustrated. Two 3 bit words are used to address a particular word. The two 3 bit input numbers are simply treated as a single six bit address. Six bits define $2^6 = 64$ bit locations. By using two 1 of 8 decoders, a total of 64 words can be addressed. The word in memory is selected by enabling each decoder with the appropriate three bit word. If the Y decoder enables Q2, Q2 will conduct and connect the bit line to the output buffer. If the decoder does not enable Q2, the output on the bit line shown will not appear at the output despite the fact the word select line may have enabled Q1.

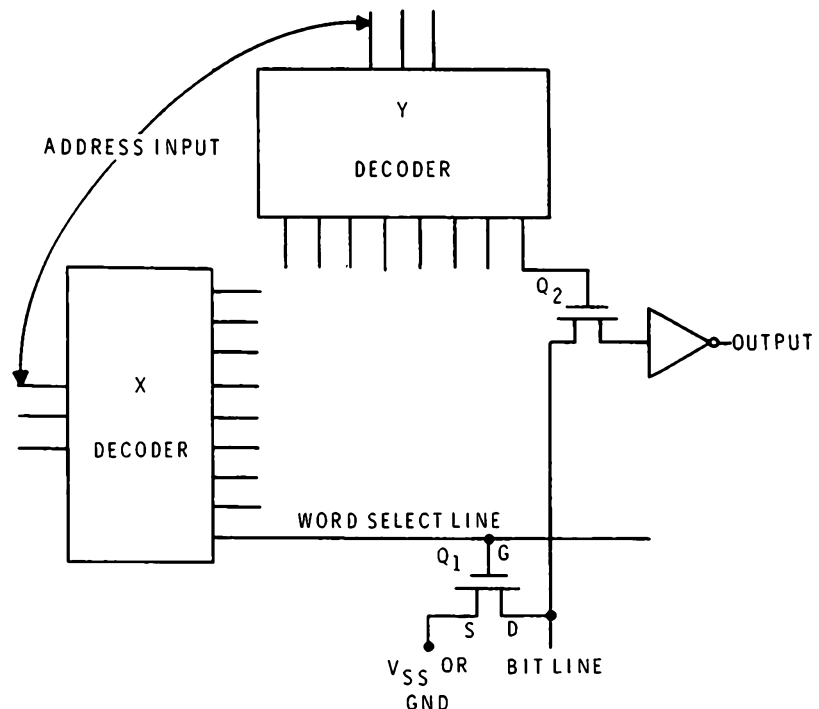


Figure 8-59
MOSFET ROM circuit.

Access Time. Like any logic circuit, a ROM has propagation delay. This means that there is a finite time between the application of an input address and the appearance of data at the output. This propagation delay is referred to as access time. This is the time it takes to find a word in the ROM and read it out. For bi-polar ROMs, this access time is usually less than 100 nanoseconds and can be as low as 20 nanoseconds. For MOS ROMs, the access time is typically several hundred nanoseconds.

ROM Applications

The ROM is extremely versatile in implementing logic functions. An appropriately programmed ROM can often be used to replace many different types of combinational circuits. It is particularly useful in replacing complex logic functions with multiple inputs and outputs. The ROM offers the advantages of faster and easier design, lower cost, smaller size, and often lower power consumption.

Combinational logic circuits generate output signals that are a function of

1. the input states
2. the types of gates used and
3. the particular unique interconnection of these gates.

The desired output states for a given set of inputs are produced by properly interconnecting the correct types of logic gates. This same logical function can be simulated by a ROM. The desired inputs are applied to the ROM address lines. These inputs specify a unique memory location. In this memory location is a binary bit pattern whose output states duplicate those produced by an equivalent combinational logic circuit. Instead of actually generating the desired output function with a logic circuit, we store the desired output states in the memory and read them out when the proper inputs appear on the address lines.

A ROM performs what is known as a table look-up function. All of the memory locations can be considered to be entries in a large table of numbers. By applying an address to the ROM, we are in effect looking up one unique number in the table. In a sense, the ROM does not perform a logic operation. The desired output states for a given set of input conditions are simply stored in the memory.

The following examples will illustrate some of the many applications of a ROM.

Random Logic. A read only memory can be used to quickly and easily implement any random logic function involving multiple inputs and multiple outputs. To design such a combinational logic circuit with standard logic gates, you first develop a truth table that defines the operation to be performed. From the truth table the Boolean equations are then written. Boolean algebra is then used to minimize these equations. From the equations, the logic circuit is developed and then implemented with standard NAND gates, NOR gates, and inverters.

When a read only memory is used, the only design step is the implementation of the truth table. The truth table defines the inputs and outputs. This is all of the information that is necessary to develop a read only memory that will perform the desired logic function. The input logic states are assigned as addresses of a read only memory. In the memory locations corresponding to the addresses are stored binary words that cause the output lines to assume the desired states with the given input address. By using a ROM you can go from truth table to finished logic circuit in one simple step. Design time is reduced considerably.

If the function to be implemented involves only a few inputs and a few outputs, such a circuit is best implemented by conventional means with logic gates. However, if the number of inputs and outputs is four or more, the use of a ROM becomes practical. Since a ROM costs more than standard SSI logic circuits, it is not practical or economical to use a ROM where very simple functions must be implemented. Four inputs and four outputs are generally regarded as the decision point between a read only memory vs. a conventional logic circuit.

Code Conversion. As we indicated earlier, code conversion refers to any multi-input/multi-output combinational logic circuit. A code converter is nothing more than a special application of such a logic circuit. Since read only memories can readily replace multi-input/multi-output combinational logic circuits, a ROM provides a simple and low cost means of code conversion.

To use a ROM as a code converter, the input code is made equal to the binary address code in the ROM. In the memory location specified by the input or address code is the desired output code. No complex logic functions need be implemented to achieve this result. The desired output codes are simply stored in the memory locations and are read out when the equivalent input code is applied. All of the most commonly used code conversion processes mentioned earlier have been implemented with read only memories.

Arithmetic Operations. Arithmetic operations are some of the most difficult functions to implement with digital circuitry. Simple combinational logic circuits have been developed to perform additions and subtractions. Various algorithms have been developed for using addition and subtraction along with other digital operations such as shifting to perform multiplication and division. More complex mathematical functions such as the trigonometric and logarithmic functions are even more difficult to implement. The read only memory provides a very simple and direct method of implementing the more complex arithmetic operations.

The multiplication of two binary numbers requires a significant amount of logic circuitry. While there are numerous methods for carrying out multiplication, all of them require an extensive amount of circuitry. Multiplication can be performed with a read only memory without the need for complex circuitry or high cost. In addition, the ROM can provide this function at lower cost, in a smaller space, and at a significantly higher speed.

The truth table shown in Figure 8-60 illustrates the concept of multiplying two binary numbers using a ROM. To simplify the explanation we will use only two bit binary numbers. Multiplying two binary numbers produces a product whose length is twice that of one of the input numbers. The two bit binary numbers serving as the multiplier and multiplicand will form a four bit product. The two input numbers are grouped so that they form a single four bit binary input number which serves as the address input to the ROM. At the address formed by the input numbers, the correct four bit product corresponding to the two bit numbers is stored. When any combination of the two bit input numbers appears on the ROM address lines, the correct product is read out. For example when one input is 10 (2) and the other is 11 (3), the input address formed is 1011. At this location is the number 0110 (6) which is the product of 2 and 3. By using a larger ROM, numbers requiring more bits can be used.

A ROM is particularly useful in handling complex mathematical operations such as the trigonometric and logarithmic functions. Instead of having the digital circuitry actually compute the sine, cosine, or tangent of a number, the ROM simply stores the trigonometric functions corresponding to the angles. In the same way, a ROM can be used to store the logarithms of specified input numbers. In these applications, the ROM is virtually a log table or trig function table. The desired angle or input number is applied to or assigned a binary address that is applied to the ROM. At the address representing the desired input angle or number is stored the correct trigonometric function or logarithm.

INPUTS				OUTPUTS			
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Figure 8-60

Truth table for
2 bit ROM multiplier.

Microprogramming

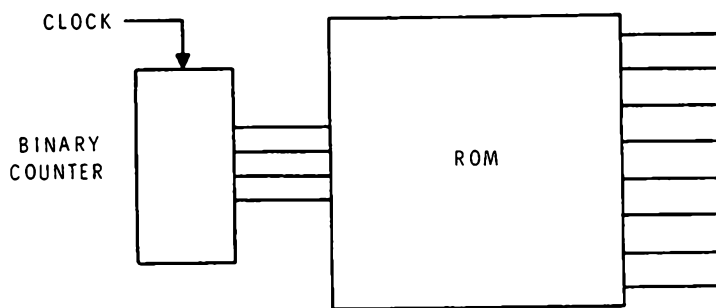
Microprogramming is a technique originally developed to systematize the automatic control logic in a digital computer. The heart of a microprogrammed control unit is a read only memory. The read only memory is combined with other logic elements to perform sequential logic operations. This combination is called a microprogrammed controller.

Most sequential operations are carried out by counters and shift registers in combination with combinational logic circuits. These circuits are used to generate a sequence of timing pulses that will control the operations in other parts of the digital system. The signals generated may increment counters, cause data transfers to take place between registers, enable or inhibit various logic gates, select a multiplexer channel or permit a decoding operation to take place. All of these operations will be timed so that they occur in the correct sequence to perform the desired operation.

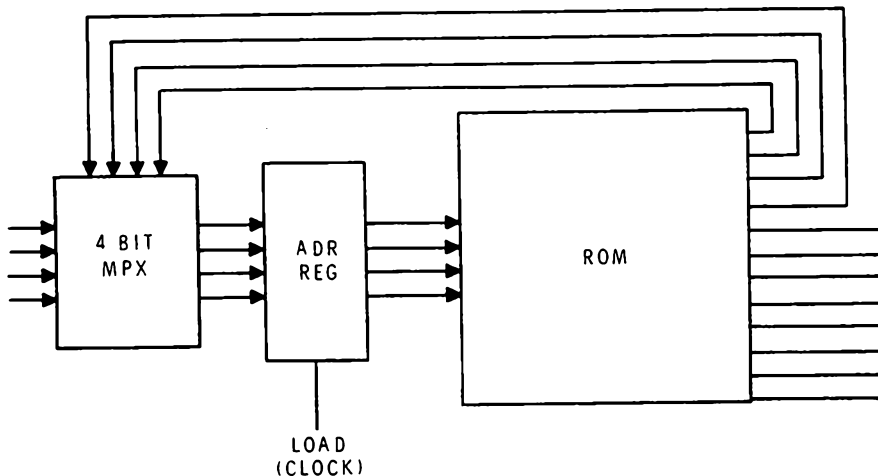
In large digital systems the control logic circuitry can become very complex. This is particularly true of the control logic in a digital computer. By the use of a microprogrammed controller, the entire network of sequential and combinational logic circuits can be replaced by a very simple circuit containing a read only memory. Figure 8-61 shows several methods of implementing sequential logic functions with a ROM.

In Figure 8-61A, the ROM is driven by a binary counter. A periodic clock signal increments the four bit binary counter. The counter output is used as the ROM address input. The address decoder is assumed to be part of the ROM itself. The ROM output consists of parallel 8 bit words. Since there are four input bits, the ROM therefore contains sixteen 8 bit words. As the binary counter is incremented, a sequence of 8 bit words appears at the ROM output. The words stored in the ROM are programmed such that the binary states appearing at the ROM outputs will cause the desired logic operations to take place in the correct sequence. Here the eight output lines can be used for a variety of control purposes. The states of these outputs are strictly a function of the bits stored in the ROM. The rate of change of the ROM outputs is a function of the frequency of the clock pulses stepping the binary counter.

A more sophisticated version of this same circuit is shown in Figure 8-61B. Again, a ROM is the main circuit element with the desired output states specified by the ROM contents. Note, however, that four of the ROM output bits are fed back around to the inputs of a 4-bit multiplexer. Another group of four input bits is applied to the multiplexer as well. The multiplexer can select either of the two 4-bit sources as an address and feed them to an address register. The address register in turn selects a specific ROM word. In this circuit, the four bits define sixteen words in the ROM. The output is a 12-bit word, eight bits for controlling external operations and four bits which are used to determine the next address of the word in the ROM.



(A)



(B)

Figure 8-61
Microprogrammed controllers using a ROM.

To operate this circuit, a four bit starting address is applied to the multiplexer from an external source. This is applied to the address register. One specific word in the ROM is addressed and its outputs appear. At this point the state of the multiplexer is changed so that the next input to the address register will come from the four bits in the current ROM output word. This permits the ROM to select the next word that should appear at the output. By repeatedly loading the address register with a clock signal, the ROM addresses are sequenced and a desired pattern of output pulses is produced. With this arrangement, the ROM words addressed can be either sequential or in any desired order. The four bit address output from the ROM could specify the next location in sequence. Alternately, it can select any other word in memory. All of this is determined beforehand in the design of the circuitry. Once the proper sequence of operations is determined, the contents of the ROM can be specified.

The term microprogramming is applicable to these circuits in the sense that the words stored in the ROM make up a specific program for carrying out a specific function. Each word in the ROM is referred to as a microinstruction. The bits of that word appearing at the ROM outputs cause certain operations to occur; in other words, that word instructs the external circuitry in the function to be performed. Each word or microinstruction stored in the memory makes up a microprogram. The microprogram defines the complete operation to be initiated. The circuits given in Figure 8-61 are only a few of many different ways that microprogrammed operations can be implemented.

Self Test Review

32. The term RAM generally refers to a:
 - a. read/write memory.
 - b. read only memory.
 - c. either of the above.
33. The binary input word to a ROM (or RAM) is often referred to as a(n) _____.
34. A 1024 bit ROM is organized to store 4 bit words. This ROM contains _____ words and requires a(n) _____ bit input address.
35. Data is written into a ROM when it is:
 - a. operating
 - b. addressed
 - c. manufactured
 - d. Data is never written into a ROM.

36. Refer to Figure 8-54. Write the binary contents of each memory location in the spaces provided.

Address	Binary Data
---------	-------------

0	
---	--

1	
---	--

2	
---	--

3	
---	--

4	
---	--

5	
---	--

6	
---	--

7	
---	--

37. The total bit capacity of the ROM in Figure 8-54 is _____ .
38. What is the word size of a 16×4 ROM?
- 4 bits
 - 8 bits
 - 16 bits
 - 64 bits
39. The propagation delay in a ROM is called _____ .
40. On a given size silicon chip which type of device will produce the largest memory?
- bipolar
 - MOS
41. A ROM can be used as a BCD to seven segment decoder. Such a ROM will have (how many?) _____ inputs, _____ outputs and a total bit capacity of _____ . The word organization is _____ x _____ .
42. A ROM could be used to perform a square root operation.
- True
 - False
43. The main logic element in a microprogrammed controller is a _____ .
44. Another name for the words stored in a microprogrammed ROM is _____ .
45. Microprogrammed controllers are
- combinational circuits.
 - sequential circuits.

Answers

32. a. Read/write memory

33. address

34. 256, 8. A 1024 bit memory organized into 4 bit words contains $1024 \div 4 = 256$ words. It takes an 8 bit address to locate any one of the words ($2^8 = 256$).

35. c. manufactured. When a programmable ROM is used, the user can store data into it once as the storage is usually permanent.

36.	Address	Binary Data
	0	0010
	1	1001
	2	1111
	3	0101
	4	0100
	5	1010
	6	0011
	7	0000

37. 32. There are 8 four bit words. ($8 \times 4 = 32$ bits)

38. a. 4 bits. In the designation 16×4 , the second number, usually the smaller of the two, refers to the word size. The first number refers to the number of words in the memory.

39. access time

40. b. MOS

41. 4 inputs, 7 outputs, 70 bits, 10×7 organization. A BCD to 7 segment decoder will have 4 inputs (the BCD input code) and 7 outputs (one for each segment). The BCD code will address 10 memory locations, one for each of the digits 0 through 9. The word in each location will have 7 bits. Therefore, the total bit capacity is $10 \times 7 = 70$. The organization of course is 10×7 .

42. a. True. A ROM can be used for square root operations. The number whose square root is to be found is applied as a binary address to the ROM. In the corresponding memory location will be the binary number representing the square root of the input.

43. ROM

44. microinstructions

45. b. sequential circuits

PROGRAMMABLE LOGIC ARRAYS

A programmable logic array (PLA) is an integrated circuit logic network that can be used to perform many different types of combinational logic functions. It offers the digital designer an alternative to the use of combinational logic circuits made with standard SSI and MSI ICs or read only memories. For many applications the PLA offers a significant improvement in performance over both conventional logic circuit implementation and read only memories.

Basically the PLA is a bipolar or MOS logic network that can be programmed during manufacturing to produce a wide variety of combinational logic functions. It is capable of translating any input code into any output code. The circuit is designed to generate a large number of sums of partial products.

Figure 8-62 shows a general logic diagram of a PLA. The multiple binary inputs (I_1 through I_{14}) are applied to inverters which are used to generate both the normal and complement versions of the input signals. The inverter outputs may then be interconnected to one of many AND gates. In one particular commercial PLA, a total of 96 twelve input AND gates are provided. These AND gates generate the product terms of the input variables. Up to 14 different input variables may be handled by this particular PLA. The products or partial products of the inputs formed by the AND gates are then connected to OR gates to form the output sums (F_1 through F_8). The selection of which input variables are applied to which AND gates and the

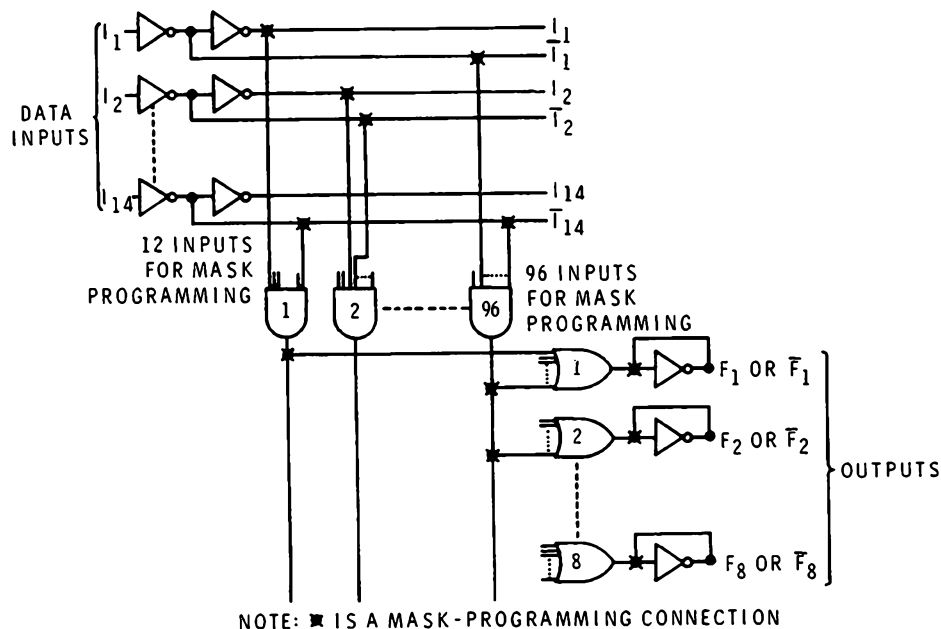


Figure 8-62
General logic diagram of
a programmable logic array.

choice of which AND gate outputs are connected to the eight available output OR gates is determined during the manufacturing of the device. By properly designing the mask that specifies the interconnections of the logic gates on the chip, a huge number of circuit configurations are possible. Note also in Figure 8-62 that even the use of an inverter on the outputs of the OR gates is programmable.

To design a logic circuit with a PLA involves basically the same procedure as developing the logic for any combinational logic circuit or selecting a ROM. The procedure generally starts with the truth table that defines the output states for each combination of input states. The output equations are written in sum-of-products form. The required product terms are listed and these are converted into the appropriate mask programming instructions for making the IC.

The PLA is particularly valuable in implementing large complex combinational logic circuits. Simple functions are readily implemented with SSI logic gates. More complex functions can be handled by one of the many available MSI functions. But when there are many input variables and many output variables, the use of standard SSI and MSI packages also becomes complex and cumbersome. The PLA can be used to generate the desired complex function and house it in a single integrated circuit package.

The PLA also offers numerous advantages over the read only memory for implementing some complex logic functions. Of course, a ROM can be used to handle logic functions involving any number of inputs or any number of outputs. However, the read only memory becomes very inefficient when all possible combinations of the input variables are not used. For example, a four input logic circuit has 16 possible different combinations. The design application may only call for the use of nine of these. The four bits on input to a ROM to be used in implementing the function define 16 memory locations, seven of which would not be used. Despite the fact the seven locations would not be used they are still present in the device and are essentially wasted. However, by using a PLA, the same logic function can be implemented more economically. PLAs offer the digital designer another option in implementing combinational logic circuits. For large, complex logic functions involving four or more inputs and outputs, it offers advantages over SSI and MSI combinational circuits and ROMs for some applications.

Self Test Review

46. The logic function performed by a PLA is determined during manufacturing.
- a. True
 - b. False
47. A PLA could be used to perform code conversion.
- a. True
 - b. False
48. A PLA is an alternative to what other types of logic circuits?
Check all that apply.
- a. Sequential
 - b. MSI functional combinational
 - c. SSI combinations
 - d. ROMs
49. The logic output equation of a PLA is in the _____
of _____ form.
50. PLAs are used primarily in implementing small simple logic functions.
- a. True
 - b. False

Answers

- 46. a. True
- 47. a. True
- 48. b, c, d. A PLA is a combinational circuit that can replace SSI, MSI combinational circuits and ROMs in large complex applications.
- 49. sum-of-products
- 50. b. False

EXPERIMENT 18

Decoders

OBJECTIVE: To demonstrate the operation of a decoder.

Materials Required

Heathkit Digital Design Experimenter ET-3200

- 1 – 74LS00 IC (443-728)
- 1 – 74LS04 IC (443-755)
- 1 – 74LS20 IC (443-798)
- 1 – 74LS42 IC (443-807)

Procedure

- Construct the decoder circuit shown in Figure 8-63. The data switches SW1-SW4 will provide the input. LED indicator L4 is the output. Be sure to connect +5 volts and ground to pin 14 and pin 7 of the two ICs.

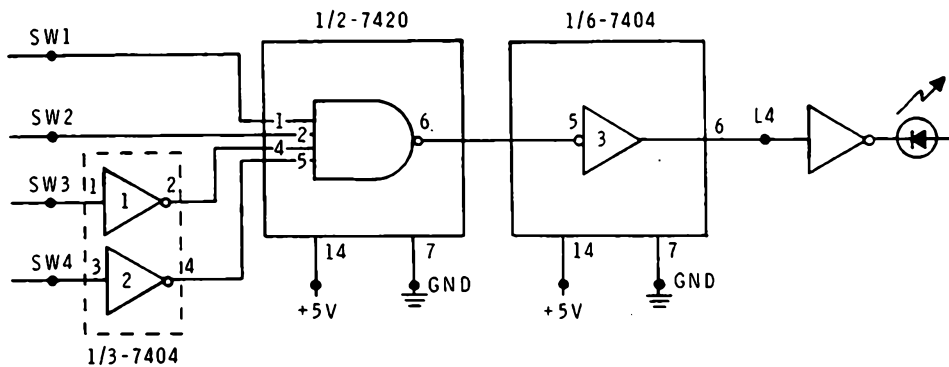


Figure 8-63
Decoder circuit for Steps 1 and 2.

- Apply the 16 states 0000 through 1111 to the circuit and observe the output. Record the binary input state where L4 lights.

Assume SW4 is the LSB. The decimal equivalent is:

3. Construct the circuit shown in Figure 8-64. Take your time in constructing this circuit to avoid wiring errors. The circuit input will come from SW3 and SW4 (LSB). You will observe the outputs on LED indicators L1 through L4. What type of circuit is this? _____

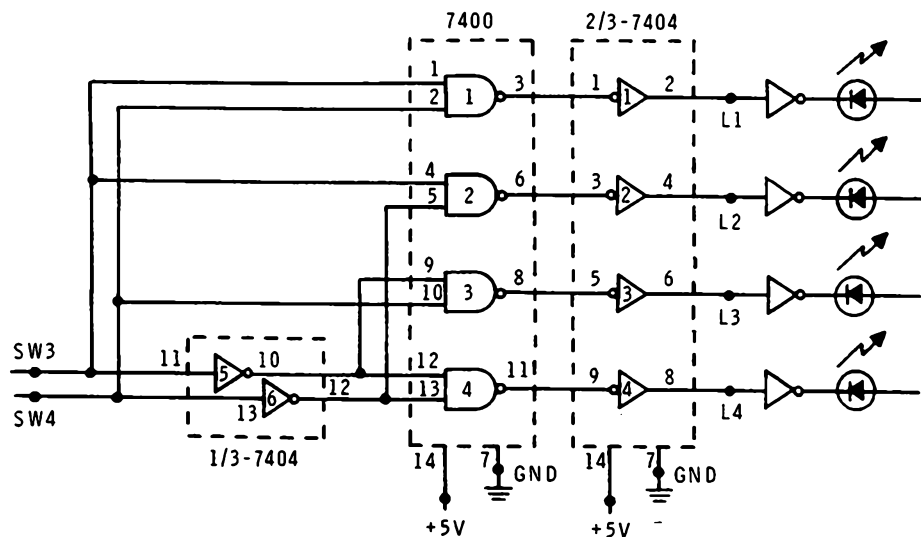


Figure 8-64

4. With SW3 and SW4, apply the inputs indicated in Table I. Record the corresponding output states of L1, L2, L3, and L4 for each of input states.

TABLE I

INPUTS		OUTPUTS (STEP 4)				OUTPUTS (STEP 6)			
SW3	SW4	L1	L2	L3	L4	L1	L2	L3	L4
0	0								
0	1								
1	0								
1	1								

Which of the following conditions did you observe for each set of inputs?

- All outputs low.
- All outputs high.
- One output low.
- Two outputs high.
- Two outputs low.
- One output high.

5. Remove the connections between the outputs of inverters 1, 2, 3, and 4 and L1, L2, L3, and L4. Connect L1, L2, L3, and L4 to the outputs of the 74LS00 IC gates, pins 3, 6, 8, and 11 respectively.
6. Repeat Step 4. Apply the inputs in Table I and record the output states in the appropriate places.
Which of the following output conditions did you observe for each set of inputs?
 - a. All outputs low.
 - b. All outputs high.
 - c. One output high.
 - d. One output low.
 - e. Two outputs low.
7. Compare your results from Steps 4 and 6 by observing the data in Table I. Then remove the circuit from the breadboarding socket.
8. Mount a type 74LS42 IC on the breadboarding socket. Connect pin 16 to +5 volts and pin 8 to ground. Connect the inputs to switches SW1 – SW4. Refer to Figure 8-65 and 8-66 for IC pin connections. You will monitor the outputs, one at a time with LED indicator L4.

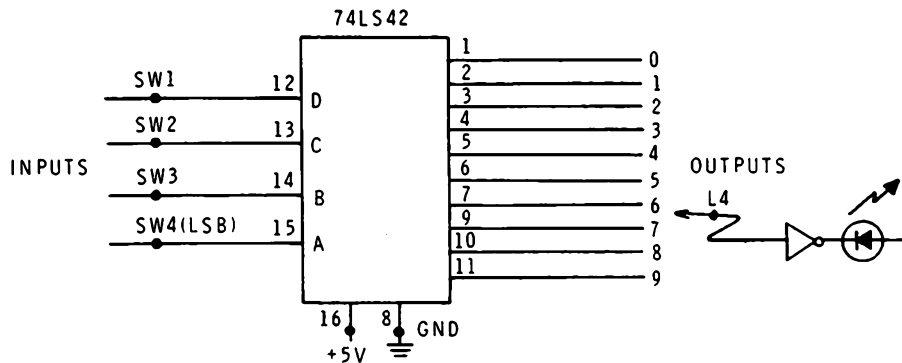


Figure 8-65
Experimental circuit
for steps 8 and 9.

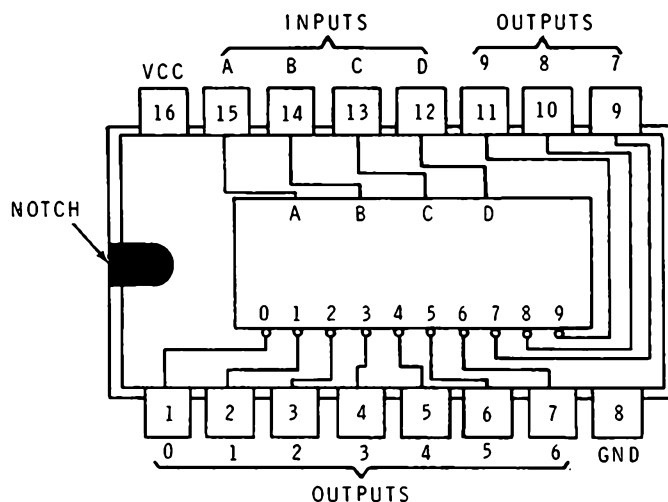


Figure 8-66
Pin connections for 74LS42
TTL IC BCD to decimal decoder.

9. Apply the inputs given in Table II. The LSB (A) is SW4. Observe the 10 outputs, one at a time, with L4 by connecting it sequentially to pins 1, 2, 3, 4, 5, 6, 7, 9, 10, and 11. Record your outputs in Table II.

TABLE II

INPUTS				OUTPUTS									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0										
0	0	0	1										
0	0	1	0										
0	0	1	1										
0	1	0	0										
0	1	0	1										
0	1	1	0										
0	1	1	1										
1	0	0	0										
1	0	0	1										
1	0	1	0										
1	0	1	1										
1	1	0	0										
1	1	0	1										
1	1	1	0										
1	1	1	1										

- a. The 74LS42 is a decimal decoder. What does this mean in terms of the outputs you observed?
- b. The 74LS42 does not recognize the six states 1010 through 1111.
 - a. True
 - b. False

Discussion

In Steps 1 and 2, you constructed and tested a simple decoder for a 4-bit input word. The 74LS20 four input NAND gate is converted into an AND gate by inverter 3 at its output. The inputs are connected so that the state 1100 (decimal 12) is decoded. When SW1 = 1, SW2 = 1, SW3 = 0, and SW4 = 0, the output will go high. For all other input codes the output will be low.

In Step 3, you constructed a one-of-four decoder circuit. The 2-bit input code comes from SW3 and SW4. You observed the four possible outputs on the LED indicators.

In Step 4 you should have found that for any set of input states, only one output is high. All others are low. This proves the one of four theory. Your data in Table I should indicate the following:

0 0, L 4 on
0 1, L 3 on
1 0, L 2 on
1 1, L 1 on

Next you removed the inverters from the outputs of the NAND gates. Indicators L1-L4 monitor the NAND outputs directly with this modification. You should have found that one output was low and the other three high for any input states. The decoder is still a one of four circuit, but the selected output is low instead of high. The data in Table I should indicate:

0 0, L 4 off
0 1, L 3 off
1 0, L 2 off
1 1, L 1 off

In comparing the one of four decoder with and without the output inverters, the output data of one should be the complement of the other as you would suspect. Both types of decoders are used depending upon the application. When the one of four outputs is high the decoder is said to have an active high output. An active low output indicates that the selected (decoded) one of four is low.

In Steps 8 and 9 you evaluated the operation of the 74LS42 BCD to decimal decoder. As you should have discovered, this circuit has active low outputs since NAND gates are used for the decoding. See Figure 8-64. Your data in Table II should correspond to the truth table in Figure 8-65. The selected output goes low. All others remain high. This circuit ignores the 1010 through 1111 states since it is an 8421 BCD (1 of 10) decoder. These six states are illegal. This is indicated by the fact when one of the six states is applied to the inputs none of the outputs go low.

EXPERIMENT 19

7 Segment Decoder-Driver and Display

OBJECTIVE: *To demonstrate the operation of an integrated circuit 7 segment decoder-driver and a 7 segment LED decimal display.*

Materials Required

Heathkit Digital Design Experimenter (ET-3200)

- 1 – 74LS90A IC (443-813)
- 1 – 74LS193 IC (443-815)
- 1 – 14495-1 IC (443-1802)
- 1 – 7-segment LED display
- 1 – 1 k Ω resistor

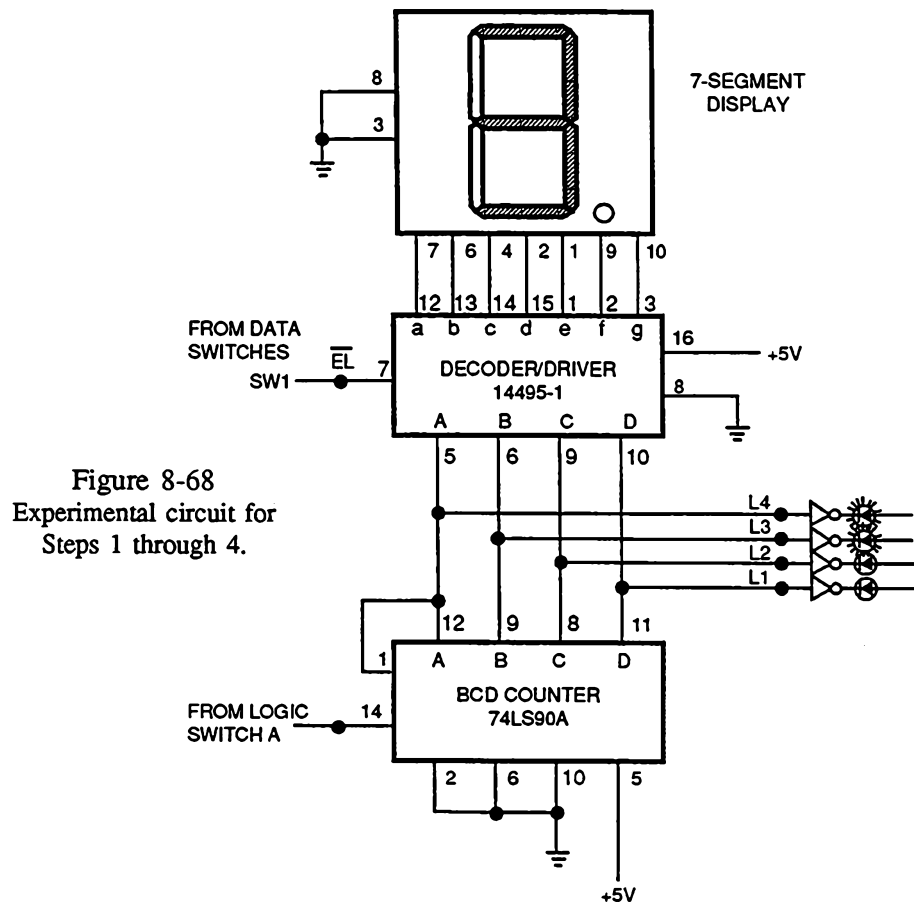


Figure 8-68
Experimental circuit for
Steps 1 through 4.

Procedure

1. Construct the circuit shown in Figure 8-68. This circuit consists of a 74LS90A BCD counter, a 14495-1 hexadecimal-to-7-segment latch/decoder ROM/driver, and a 7-segment LED display. The four BCD outputs of the 74LS90A counter drive LED indicators L1 through L4 and the decoder-driver. The output of the decoder-driver is used to drive the 7-segment display. The pin connections to the decoder-driver and the 7-segment LED display are given in Figure 8-69. As before, use care in wiring the circuit to prevent wiring errors. Don't forget to connect +5 volts and ground to each IC. The decoder driver IC is a 7-segment decoder-driver. It accepts the four-bit BCD number from the 74LS90A BCD counter. The 14495-1 IC contains a four bit latch register which can be used to store the four-bit input. The output of this latch is fed to the decoder circuit that converts the BCD input into the 7-segment output code described earlier. Driver transistors in the IC provide the current necessary to operate the 7-segments of the display. The four bit latch is loaded or enabled by the \overline{EL} input pin.

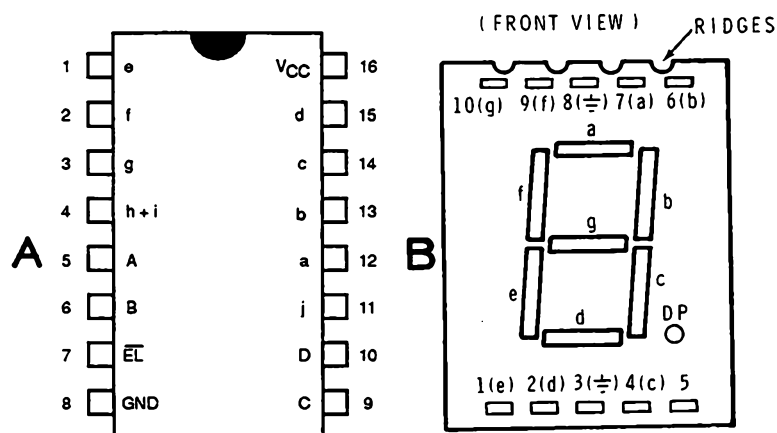


Figure 8-69
 (A) Pin connections for 14495-1
 decoder-driver IC. (B) Pin connections
 for type FND500 7-segment LED display.

2. Apply power to the circuit. Set data switch SW1 to binary 0. Step the BCD counter with the A logic switch. As you do, note the binary LED displays L1 (MSB) through L4 (LSB) and the 7-segment display. Check to see that the binary number shown is equivalent to the decimal number indicated. Step the counter through its ten states several times to see that the circuit is performing properly.
3. Remove the lead connecting pin 14 of the 74LS90A counter to the A logic switch and connect it to the CLK output. Set the clock frequency to 1 Hz. The clock will now automatically step the counter and permit you to observe both the BCD and decimal outputs of the circuit automatically.
4. When the decimal display reads 7, quickly set SW1 to the binary 1 position. Continue to observe LED indicators L1 through L4 and the 7-segment display and note your result.

5. Modify your experimental circuits so that it appears as shown in Figure 8-70. Remove the 74LS90A IC and in its place install the 74LS193 binary counter. You will continue to use the 1 Hz clock to step the counter. Data switch SW2 will be used to reset the counter. Connect all unused outputs on 74LS193 to 5V.
6. Set data switch SW2 to the binary 0 position. Check to see that SW1 is in the binary 0 position. As before, the counter should change states at a 1 Hz rate as indicated by the LED indicators L1 through L4 and the 7-segment display. While you are observing the LED displays, note the status of the 7-segment readout during the 6 invalid codes for BCD operation.

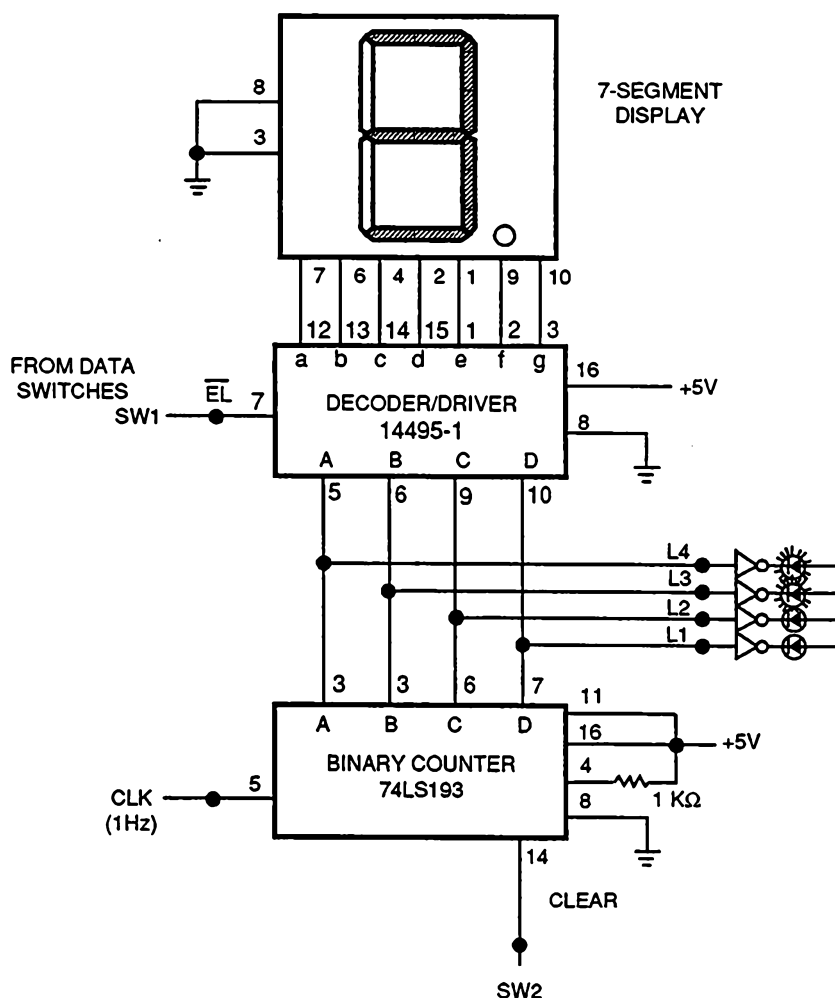


Figure 8-70
Experimental circuit for Steps 5 and 6.

7. Set data switch SW2 to the binary 0 position. Check to see that SW1 is in the binary 0 position and SW4 is in the binary 1 position. As before the counter should change states at a 1 Hz rate as indicated by the LED indicators L1 through L4 and the 7 segment display. While you are observing the LED displays, note the status of the 7 segment readout during the 6 invalid codes for BCD operation. Does the decoder-driver recognize the six 4-bit binary codes normally considered to be invalid in the BCD coding system? _____.

If your answer to the question above is yes, record the characters displayed by the 7 segment readout during these six invalid states.

1010 _____.

1011 _____.

1100 _____.

1101 _____.

1110 _____.

1111 _____.

Discussion

In this experiment you demonstrated the operation of a decoder-driver circuit that accepts a binary or BCD input code and generates the 7-segment display signals to produce the numbers 0 through 9 and other characters. In Steps 1 through 5 you used a 74LS90A BCD counter to drive the decoder-driver and display. This circuit counts in the standard 8421 BCD code. As you stepped the counter with the A logic switch, you should have generated the four-bit BCD codes as displayed by LED indicators L1 through L4. At the same time, the corresponding decimal digit should have been displayed on the 7-segment readout. Using the 1 Hz clock signal to run the circuit permitted you to observe the outputs while the circuit stepped automatically.

In Step 4 you used data switch SW1 to set the \overline{EL} input to the binary 1 state when the decimal output was 7. You should have found that the 7-segment display continued to indicate 7 while the clock continued to step the counter and display the sequential BCD states on LED indicators L1 through L4. What you did when you set SW1 to the binary 1 position was to store the number 0111 in the latch storage register of the decoder-driver. The 7-segment readout displays only the binary or BCD number stored in that internal register. By setting the \overline{EL} line high you effectively inhibited the BCD inputs from the binary counter from further affecting the decoder-driver. Of course the BCD counter continued to sequence through its normal states as indicated by the changing conditions on L1 through L4. During the previous part of the experiment, you set SW1 to binary 0 condition. This enables the latches or D flip-flops in the storage register and permitted the 7-segment outputs to follow the BCD input.

Figure 8-71 shows a simplified block diagram of the 7-segment decoder-driver/latch. The 4-bit inputs are applied to the data inputs of latches that are enabled by the \overline{EL} line. The outputs of these four flip-flops are fed through a decoder. The decoder outputs then drive a circuit that generates the 7-segment code necessary to display the digits 0 through 9 and letters A through F. The output devices are current driven transistors that supply the proper current to the segments in the driver.

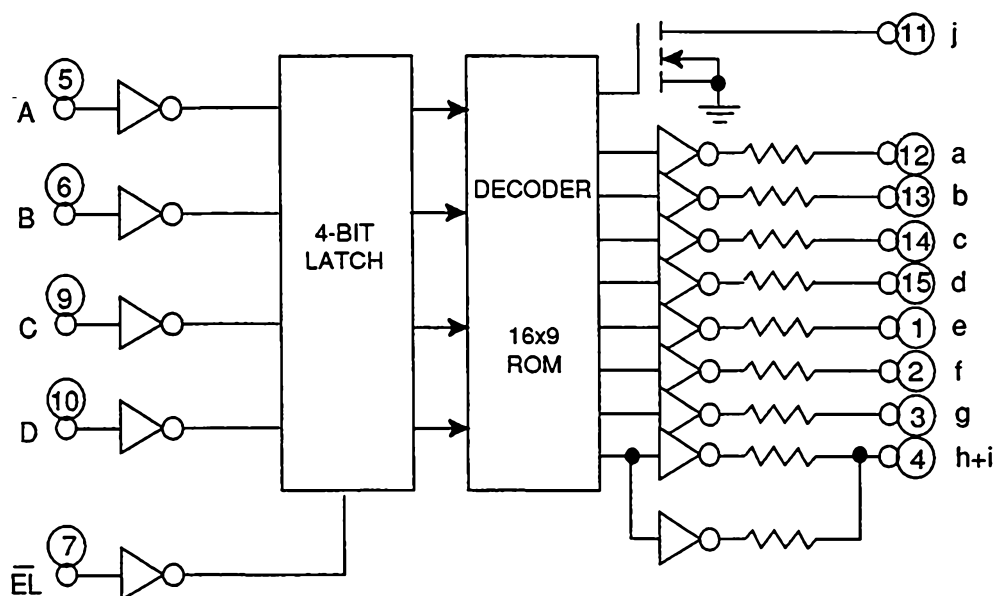


Figure 8-71
Block diagram of 14495-1
7-segment decoder driver IC.

Finally, you replaced the BCD counter with a standard 4-bit binary counter. In Step 6, you should have found that the decoder-driver does recognize the six states normally considered to be invalid in the BCD code. In these six states, the decoder-driver causes the letters A, B, C, D, E, and F to be displayed on the 7-segment display.

In step 5, you demonstrated the operation of the ripple blanking input. When you set SW4 to the binary 0 state, you effectively produced 0 suppression. When the counter stepped to the 0000 state, the 7-segment display should have been blank. It will not display a 0 when the $\overline{\text{RBI}}$ input is low.

Finally, you replaced the BCD counter with a standard 4-bit binary counter. In step 7, you should have found that the 9368 decoder-driver does recognize the six states normally considered to be invalid in the BCD code. In these six states, the decoder-driver causes the letters A, B, C, D, E, and F to be displayed on the 7-segment display.

EXPERIMENT 20

Multiplexers

OBJECTIVES: *To demonstrate the operation and application of digital multiplexers.*

Materials Required:

Heathkit Digital Design Experimenter (ET-3200)
 1 – 74LS00 IC (443-728)
 1 – 74LS151 IC (443-878)
 1 – 74LS193 IC (443-815)
 1 – 1 k Ω resistor

Procedure

1. Construct the circuit shown in Figure 8-72. Use a type 74LS00 IC, and be sure to connect +5 volts and ground to pins 14 and 7 respectively.

Study the circuit in Figure 8-72 and answer the questions below.

- a. What type of circuit is this? _____
- b. The two signal sources are _____ and _____.
- c. The control input is _____.

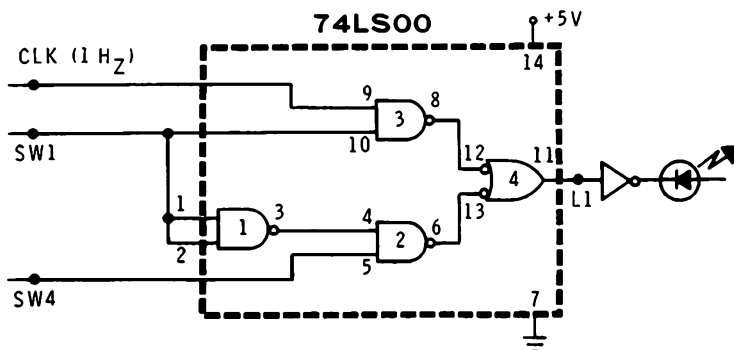


Figure 8-72
 Multiplexer circuit for Steps 1 and 2.

2. Apply power to the circuit. Set SW1 to the binary 1 position and note the circuit output on LED indicator L1. Note the effect of switching SW4 off then on.

The output is _____.

Set SW1 to the binary 0 position. Note the effect of switching SW4 off and on.

The output is _____.

Discussion For Steps 1 and 2

In step 1, you constructed a 2-input multiplexer or data selector circuit. You evaluated the operation of this circuit in step 2. The two input signal sources are the 1 Hz clock signal (CLK) and the signal from SW4. Data switch SW1 is used as the control input.

Refer to Figure 8-72. Whenever the SW1 switch is in the binary 1 position, gate 3 will be enabled. The 1 Hz CLK signal will pass through to the output and will cause LED indicator L1 to switch off and on at a 1 Hz rate. At this time, gate 1 (which is connected as an inverter) inhibits gate 2, thereby preventing SW4 from influencing the output of the circuit.

When SW1 is placed in the binary 0 position, gate 3 is inhibited. This causes the output of inverter 1 to be high, thereby enabling gate 2. At this time, the state of SW4 will be transferred to the output. So, the circuit is capable of selecting one of two data sources and routing it through to the single output. Switch SW1 controls which data source is selected. This circuit is a logical equivalent for a simple single-pole double-throw (SPDT) switch.

Procedure (Continued)

- Construct the circuit shown in Figure 8-73. This circuit uses an 8-input data selector TTL type 74LS151. The desired input is selected by a 3-bit code ABC which is derived from a 74LS193 binary counter. This 3-bit code is monitored on LED indicators L2 through L4. The counter is stepped manually by logic switch A.

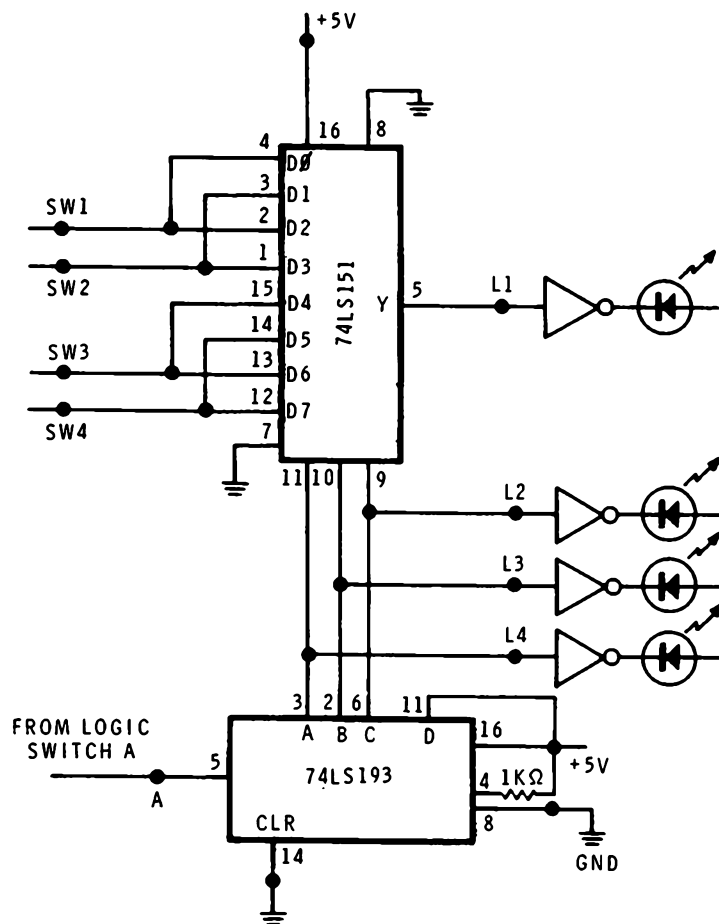


Figure 8-73
Experimental circuit for Steps 3 through 6.

The inputs to the data selector or multiplexer are derived from data switches SW1 through SW4. The data selector output will be monitored on LED indicator L1.

The pin connections for the 74LS151 data selector are given in Figure 8-74. The logic diagram for this circuit is shown in Figure 8-75.

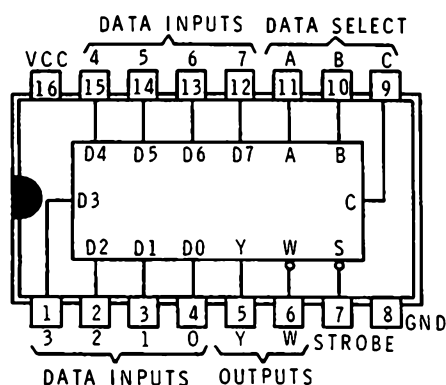


Figure 8-74
Pin connections for 74LS151
multiplexer.

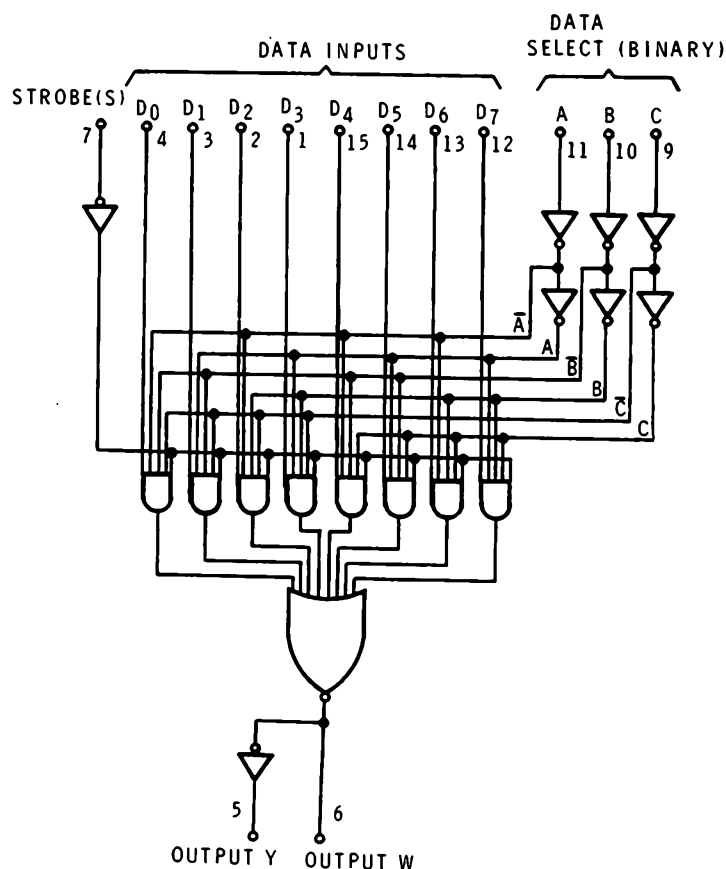


Figure 8-75
Logic diagram of 74LS151
multiplexer.

4. Refer to Figure 8-75. Write the Boolean output equation for the multiplexer circuit shown. Doing this will help you to understand what the circuit does.

Y = _____

5. Apply power to your experimental circuit. Step the binary counter with the A logic switch until the L2, L3, L4, states are 000. Observing the experimental circuit diagram in Figure 8-73 and the data selector logic diagram in Figure 8-75, determine which input on the 74LS151 is enabled with this binary code. Then operate each of the data switches SW1 through SW4 and determine which one effects the data selector output. Record this information in Table I.

Increment the binary counter with logic switch A so that the LED indicators L2 through L4 read 001. Again, determine which input (D0 through D7) of the 74LS151 multiplexer is enabled. Confirm this by actuating SW1 through SW4 until you determine which switch causes a change in the output on indicator L1. Record this information in Table I. Continue incrementing the counter for all eight states and completing the table as indicated.

TABLE I

INPUT CODE			MPX INPUT	DATA SOURCE
C	B	A		
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Discussion for Steps 3 Through 5

In these steps you used a binary counter to select one of eight inputs on the 74LS151 multiplexer. The binary code from the counter is decoded within the multiplexer and enables one of the eight lines (D0 through D7). The decimal value of the input line enabled corresponds to the equivalent binary input code. For example, with the input code 101, multiplexer input D5 is enabled.

Data switches SW1 through SW4 are used as the data source for the multiplexer inputs. Since there are only four switches, each is wired to two of the multiplexer inputs. As you can see from Figure 8-73, SW1 drives D0 and D2, SW2 drives D1 and D3, SW3 drives D4 and D6 while SW4 drives D5 and D7. Your data in Table I (MPX input) should reflect this.

In operating this circuit you should have discovered that only one of the eight inputs is enabled at a time. This permits only one input switch to effect the output. Setting the switch alternately to binary 0 and binary 1 should have caused LED indicator L1 to follow.

In Step 4 you are asked to write the Boolean equation for the 74LS151 multiplexer circuit in Figure 8-75. Your Boolean equation should appear as shown below.

$$Y = \bar{S}(\bar{A}\bar{B}\bar{C}D_0 + \bar{A}\bar{B}\bar{C}D_1 + \bar{A}\bar{B}\bar{C}D_2 + \bar{A}\bar{B}\bar{C}D_3 + \bar{A}\bar{B}\bar{C}D_4 + \bar{A}\bar{B}\bar{C}D_5 + \bar{A}\bar{B}\bar{C}D_6 + \bar{A}\bar{B}\bar{C}D_7)$$

Note that each of the inputs D0 through D7 is enabled by its own unique input code. Strobe input \bar{S} is used to enable or disable the entire circuit. When the \bar{S} input line is low, all eight input gates are enabled thereby permitting data to pass through to the output. If the strobe input is high, all of the gates are inhibited and no data will pass through to the output.

It is evident from the logic equation that the multiplexer is a sum-of-products generator for all possible combinations of the inputs A, B, and C. By selecting the desired input lines, any Boolean equation in the sum-of-products form for three variables A, B, and C can be generated. You will demonstrate this concept in the next steps.

Procedure (Continued)

6. Modify your experimental circuit to conform to Figure 8-76. Remove the connections from the multiplexer inputs to data switches SW1 through SW4. Wire the inputs of the multiplexer as shown in Figure 8-76. Connected in this way, the multiplexer becomes a serial data word generator or a Boolean function generator.

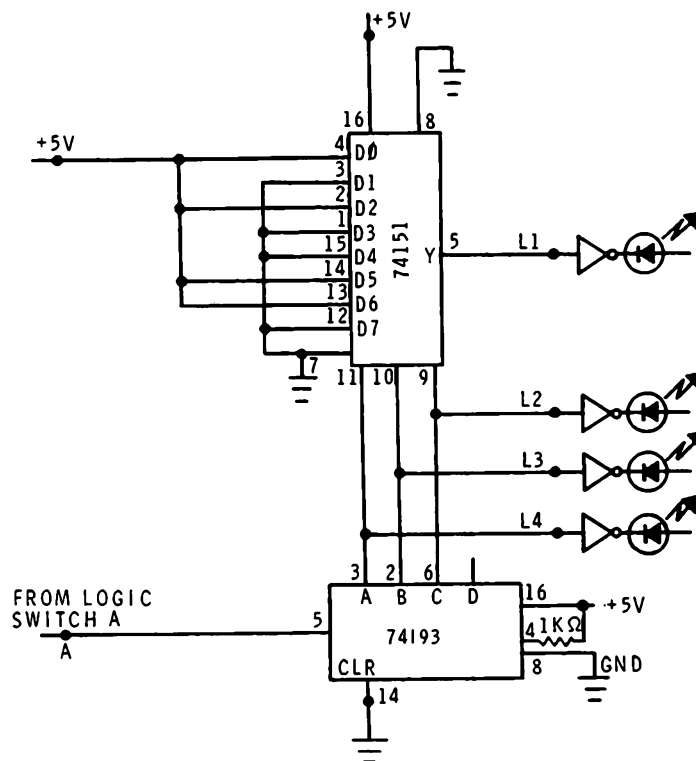


Figure 8-76
Experimental circuit for Steps 6 through 10.

7. Apply power to the circuit. Step the counter with the A logic switch until it is in the 000 state as indicated by LEDs L2 through L4. At this time observe the multiplexer output on LED indicator L1. This is the first bit of an eight bit word to be generated by the 74LS151 multiplexer. The state you are observing at this time is the LSB of the eight bit word.

Next, step the counter with the A logic switch. At each counter state, note the multiplexer output by observing L1. Increment the counter until the last bit of the word (counter state 111) is obtained. Record the binary word developed and its equivalent decimal value in the spaces provided below.

Serial output binary word = _____

Decimal equivalent = _____

8. Without changing the experimental circuit, assume that it is being used as a Boolean function generator. Step the counter through its eight states and again note the L1 output condition for each of the counter states. Use this data to complete the truth table shown in Table II.

TABLE II

INPUTS			OUTPUT
C	B	A	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

9. Using the procedure you learned in a previous unit, write the Boolean equation (sum-of-products) from the truth table in Table II. Record your Boolean equation below.

Y = _____

10. Observing the experimental circuit in Figure 8-76 and using the multiplexer logic diagram in Figure 8-75 for reference, write the output equation for the multiplexer. Note the states of the multiplexer inputs, then combining this information with the Boolean equation for the multiplexer you developed earlier, you should be able to write the sum-of-products expression of the output Y for the multiplexer input connections given here.

Y = _____

Compare the equation developed here with the equation you produced from the truth table in the previous step.

Discussion For Steps 6 Through 10

The circuit you wired in Step 6 permits the multiplexer to be used as either a serial binary word generator or a Boolean function generator. In each case the states of the multiplexer inputs determine the states of the outputs for each of the eight possible binary input codes from the counter.

In Step 7 you used the circuit to generate a serial binary word. The word produced by this circuit is 01100101. With the counter in state 000, the multiplexer was observing the LSB of this number. This is the binary state at the D_0 input. As the counter was incremented, each new bit in the serial word was generated until the 7th counter state (111) was reached. This represents the most significant bit of the word (D_7). The decimal equivalent of this binary number is 101_{10} . Another way to look at this circuit is as a parallel-to-serial converter. The parallel input number 01100101 is converted to a serial format by the 74LS151 IC.

In Step 8 you used the same circuit but interpreted its output as a Boolean function generator. Assuming the inputs to the multiplexer are the logic signals A, B, and C, the circuit output Y is a function of these inputs and the states of the multiplexer inputs D_0 through D_7 . As you saw earlier, the multiplexer is capable of generating all eight product terms specified by the three bit input. By enabling or disabling the various inputs these terms can be added to the output. In this circuit multiplexer inputs D_0 , D_2 , D_5 , and D_6 have been enabled by a +5 volt signal. This means that the terms associated with these inputs will appear in the output. You verified this by plotting a truth table for the output. You should have found that the circuit produced a binary 1 output when input states 000, 010, 101, and 110 occurred.

To determine the Boolean equation, you observed the truth table and wrote down a product term developed from the input states where a binary 1 appears in the output. For example, an output appears when input state ABC is equal to 010. You would then write a product term equal to $\overline{A}B\overline{C}$. All of the terms are then summed on the output equation. Your equation should be:

$$Y = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + \overline{A}BC$$

In Step 10 you analyzed the connections to the 74LS151 multiplexer and wrote the Boolean equation from the circuit connections. Referring to the Boolean equation you wrote for the multiplexer circuit itself, you should have found your Boolean equation to be the same as that developed from the truth table.

EXPERIMENT 21

Exclusive OR

OBJECTIVES: To demonstrate the operation of exclusive OR and exclusive NOR gates.

Materials Required:

Heathkit Digital Design Experimenter (ET-3200)

1 – 74LS00 IC (443-728)

1 – 74LS86 IC (443-891)

1 – 4001 IC (443-695)

Procedure

1. Wire the circuit shown in Figure 8-77. You will use a type 74LS00 IC. The inputs to the circuit A and B will come from data switches SW1 and SW2. The output C will be indicated on L1. Be sure to connect +5 volts to pin 14 and ground to pin 7 on the IC.

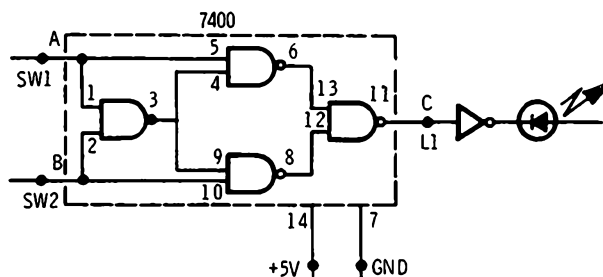


Figure 8-77
Experimental circuit
for Steps 1 through 3.

TABLE I

A	B	C
0	0	
0	1	
1	0	
1	1	

2. Apply power to the circuit. Using data switches SW1 and SW2, apply the four separate sets of inputs indicated in Table I. For each set of inputs, record the corresponding output C and complete Table I.
3. From the truth table you completed in Table I, write the output equation for the circuit you evaluated and record below.
C = _____
From the truth table and the equation you can see that the circuit does perform the _____ logic function.

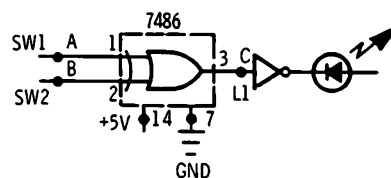


Figure 8-78
Experimental circuit for Steps 4 and 5.

4. Mount a type 74LS86 IC on the breadboarding socket. Wire it as shown in Figure 8-78. The pin connections for this IC are shown in Figure 8-79. As in the earlier steps, data switches SW1 and SW2 will be used to supply the inputs to the circuit. You will monitor the output on LED indicator L1.

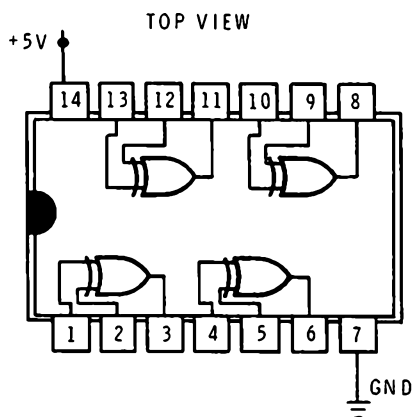


Figure 8-79
Pin connections for type 74LS86 IC.

5. Apply the input states A and B as indicated in Table II to the experimental circuit. For each set of inputs, monitor the output C and record the state in the appropriate space in Table II. Study the truth table and determine what function the circuit is performing. _____

TABLE II

A	B	C
0	0	
0	1	
1	0	
1	1	

6. Construct the circuit shown in Figure 8-80. The inputs will come from the data switches and you will observe the outputs on the LED indicators. Switch SW4 and LED indicator L4 can be considered to be the LSB of the four bit binary word input and output.

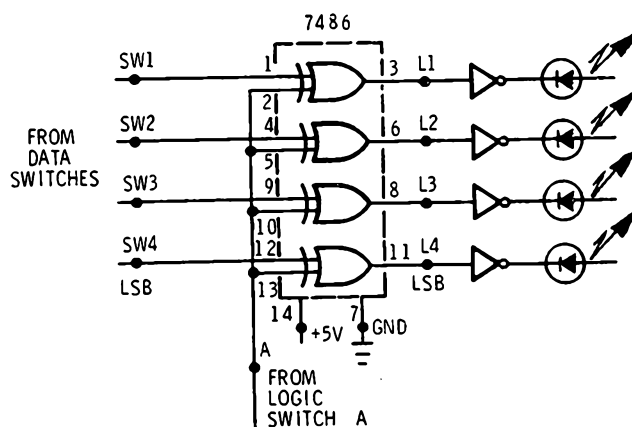


Figure 8-80
Experimental circuit
for Steps 6, 7, and 8.

7. Set all of the input data switches to the binary 0 state. Observe the output state and record your output value below and in Table III.

Input = 0000, Output = _____

Next, depress the A logic switch. Again observe the output indicators and record the state presented below and in Table III.

Input = 0000, Output = _____

For each of the input states recorded in Table III, record the output states with the A logic switch in its normal position and in its depressed position.

8. Study the results in Table III. Using this information and the circuit in Figure 8-80 plus the results of your previous steps, determine the function of this circuit.

TABLE III

INPUTS				OUTPUTS							
SW1	SW2	SW3	SW4	A NORMAL				A DEPRESSED			
				L1	L2	L3	L4	L1	L2	L3	L4
0	0	0	0								
1	1	1	1								
1	0	1	0								
0	0	1	1								

Discussion For Steps 1 Through 8

In Step 1 you constructed an exclusive OR circuit using a 74LS00 quad two input NAND gate. Applying the inputs A and B indicated in Table I you should have found that the output of the circuit was binary 1 when either one but not both of the inputs were binary 1. As long as the inputs were complementary the output was binary 1. For equal value binary inputs, the output was binary 0. This is typical of the exclusive OR function. In writing the Boolean equation of this circuit from the truth table you should have found it to be $C = \bar{A}B + A\bar{B}$.

In Step 4 you wired a circuit using a 74LS86 IC. This is a quad exclusive OR gate. Four complete exclusive OR circuits are contained within this package. With an MSI device such as this, it is unnecessary to implement exclusive OR gates with gate packages as you did in Step 1.

Applying the inputs given in Table II you should have found that the circuit in the 74LS86 does indeed perform the exclusive OR function. The results you obtained in Table II should be equivalent to that you obtained in Table I.

The circuit you constructed in Step 6 not only illustrates the operation of an exclusive OR gate, but also shows one practical application. Here a four bit binary word from the data switches is applied to the exclusive OR gates. You noticed from Figure 8-80 that one input of each of the exclusive ORs is connected together and wired to logic switch A. With logic switch A in the normal nondepressed state, output A is binary 0. Knowing the operation of the exclusive OR and the state of the input data switches you should be able to determine what the exclusive OR gate outputs are. Referring to the truth table for an exclusive OR you can see that when one input to each of the exclusive OR gates is binary 0, the output of that exclusive OR gate will be equal to the binary state of the other input.

When logic switch A is depressed, a binary 1 is applied to all of the exclusive OR gates. Looking at the truth table then you can see that the output of the exclusive OR will be the complement of its other input. The circuit in Figure 8-80 therefore is a binary complements. Under the control of the A input, the output can be equal to the input or its complement depending upon the state of the A input. In Table III you applied four different inputs and monitored the outputs with the A logic switch in both the normal and the depressed states. With the A switch in the normal position, the LED indicators should indicate the state of the inputs. With the A logic switch depressed, the LED indicators should show the complement of the input states.

Procedure (Continued)

9. Construct the circuit shown in Figure 8-81. You will use a type 4001 CMOS quad NOR gate. Inputs A and B will come from data switches SW1 and SW2 as before. You will observe the output C on LED indicator L1.

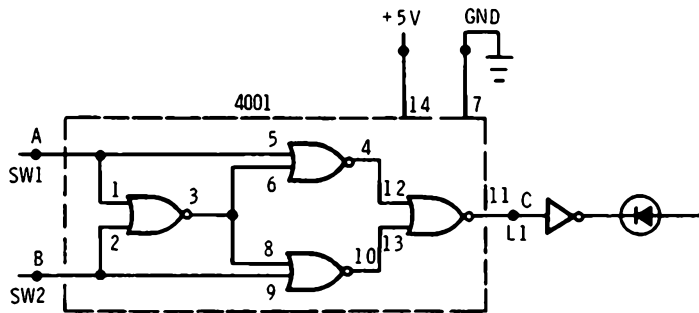


Figure 8-81
Experimental circuit
for Steps 9, 10, and 11.

10. Apply the input states shown in Table IV. Record the output C for each of the corresponding input states. Study the table and determine what function the circuit is performing.

From Table IV, write the Boolean equation expressing the operation of this circuit.

C = _____

TABLE IV

A	B	C
0	0	
0	1	
1	0	
1	1	

11. Study the circuit in Figure 8-81. Write the output expression C for the circuit in terms of inputs A and B. Use Boolean algebra and DeMorgan's theorem to manipulate the output expression into its simplest form. Studying the results of your equation, determine the function of the circuit. This circuit is known as a _____ or a _____.

Discussion For Steps 9, 10, and 11

The circuit you constructed in Step 9 is an exclusive NOR gate. If you write the output equation of the circuit you will find that it is $C = A B + \overline{A} \overline{B}$. This, of course, is the Boolean equation for an exclusive NOR gate. The exclusive NOR function is the complement of the exclusive OR function.

You plotted a truth table to verify the operation of this circuit. You should have found that the circuit output C was equal to binary 1 when the two inputs were equal to one another. When the inputs were complementary, the outputs were 0. Another name for the exclusive NOR is comparator or equivalence.

EXPERIMENT 22

Exclusive OR Applications

OBJECTIVES: *To demonstrate practical applications of exclusive OR and exclusive NOR gates.*

Introduction

In this experiment you are going to show some of the ways in which exclusive OR and exclusive NOR gates are used to perform practical logic operations. The circuits that you will build and experiment with represent the most popular applications for the exclusive OR gate. But perhaps more important is the fact that in this experiment you begin to combine both combinational and sequential logic circuits to perform more sophisticated operations. This experiment, therefore, is a vital step in your increased understanding of complex digital logic circuits.

Before you perform each of the experiments to be described, study the experimental circuit before you construct it. Be sure that you understand what types of components are being used and how they are interconnected. Determine the operation of the circuit so that you will know what to look for when you evaluate its performance.

Each of the circuits that you will experiment with here involves several integrated circuits. There is a substantial amount of wiring involved. Our previous precautions about wiring mistakes are even more important in this experiment. Take your time when interconnecting the circuits to avoid wiring errors. If you should have difficulty in making the experiment perform, verify the circuit wiring first.

Materials Required:

Heathkit Digital Design Experimenter ET-3200

- 1 — 74LS00 IC (443-728)
- 1 — 74LS20 IC (443-798)
- 1 — 74LS04 IC (443-755)
- 1 — 74LS151 IC (443-878)
- 2 — 74LS76 IC (443-829)
- 1 — 74LS86 IC (443-891)
- 1 — 74LS193 IC (443-815)
- 1 — 74LS95 IC (443-814)
- 1 — 1 k Ω resistor
- 1 — DC Voltmeter or logic probe

Procedure

- Construct the circuit shown in Figure 8-82. This is a four bit parity generator circuit made from the 74LS86 quad exclusive OR IC. The input to the parity generator circuit will come from the binary counter 74LS193. You will step the counter through its sixteen states and observe the parity bit output on your DC voltmeter or logic probe.

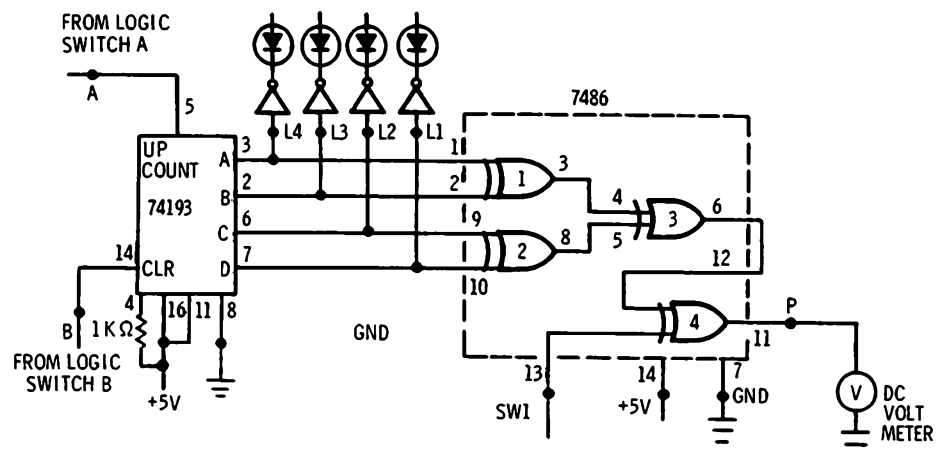


Figure 8-82
Parity generator circuit
for Steps 1 through 5.

- Study the circuit shown in Figure 8-82. Assume SW1 is in the binary 0 position. Will the parity output bit P be odd or even?

With SW1 in the binary 1 position, will the parity output bit P be odd or even? _____

3. Apply power to the circuit. Use logic switch B to clear the binary counter. Set SW1 to binary 0 position. Connect your DC voltmeter to pin 11 of the 74LS86 IC. Record the binary output state of P with the input code 0000. Use the appropriate column in Table I. (For TTL circuits binary 1 = +3.5 volts, binary 0 = +0.1 volts)

Increment the binary counter with the A logic switch. Observe the counter states on the LED indicators. For each of the sixteen input states, record the corresponding binary output state of P in Table I.

TABLE I

D	C	B	A	P(SW1=0)	P(SW1=1)
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

4. Repeat Step 3 with data switch SW1 in the binary 1 position. Record your output states in the appropriate column of Table I. Study your results in Table I and determine which position of SW1 produces odd and even parity. What function does X-OR gate 4 in Figure 8-82 play in determining the nature of the output parity bit? _____
5. Assume that the circuit in Figure 8-82 is a parity checker circuit rather than a parity generator. X-OR gates 1 through 3 form the parity generator circuit while X-OR gate 4 then becomes the comparator to compare the parity bit transmitted with the data (simulated by SW1) and the parity bit generated by the word itself through gates 1, 2, and 3. Answer the following questions:
- What indication does the output P give if this is a parity checker circuit? _____
 - Does the parity generator circuit made up of gates 1, 2, and 3 generate odd or even parity? _____
 - Assume that the input word from the binary counter is 1010 and the input parity bit from SW1 is binary 1. Does a parity error exist? _____

Discussion For Steps 1 Through 5

The circuit you constructed in Figure 8-82 is a parity generator. X-OR gates 1, 2, and 3 form an even parity generator circuit. X-OR gate 4 is used as a complements to provide either odd or even parity output. When SW1 is in the binary 0 position, the output of X-OR gate 4 will be an even parity bit. Setting SW1 to the binary 1 position causes the even parity output circuit from X-OR gate 3 to become complemented and thus produce an odd parity output.

You used the 74LS193 binary counter as a truth table generator for the circuit. Rather than using the data switches as an input, you use the binary counter since the four bit binary code is generated automatically each time you increment the counter with logic switch A. The binary counter makes an excellent truth table generator for producing all possible combinations of four logic signals to apply to a combinational logic circuit for purposes of determining its output states. You recorded in Table I both the odd and even parity output bits by setting SW1 to binary 0 and then the binary 1 state. With SW1 in the binary 0 position even parity is produced. Even parity means that the total number of ones in the binary word including the parity bit is even. With SW1 in the binary 1 position, odd parity is generated. Here the total number of ones in the input word including the parity bit is odd. Note that the odd and even parity output bits are complementary.

When using the circuit in Figure 8-82 as a parity checker circuit, X-OR gates 1, 2, and 3 form an even parity generator circuit. The output of X-OR gate 3 is the even parity bit which is applied to one input of X-OR gate 4. This is compared to an externally transmitted parity bit which in this case is represented by SW1.

If the binary input code is 1010, the even parity generator will produce the binary 0 output bit so that even parity is produced. At the same time if SW1 represents a parity input bit which is binary 1, a parity error will be generated. The parity error comparator or detector is X-OR gate 4. A binary 1 output will be produced if the two inputs to this gate are complementary. This indicates a parity error. No parity error is indicated when the two bits are equal or alike.

Procedure (Continued)

6. Wire the circuit shown in Figure 8-83. You will again use the 74LS193 binary counter as a truth table generator. The binary code will be applied to the circuit made up from the 74LS86 quad exclusive OR IC. You will observe the output on LED indicators L1 through L4. This is a code converter circuit that will change the binary code from the 74LS193 counter into another code. You will determine that code in the next step.

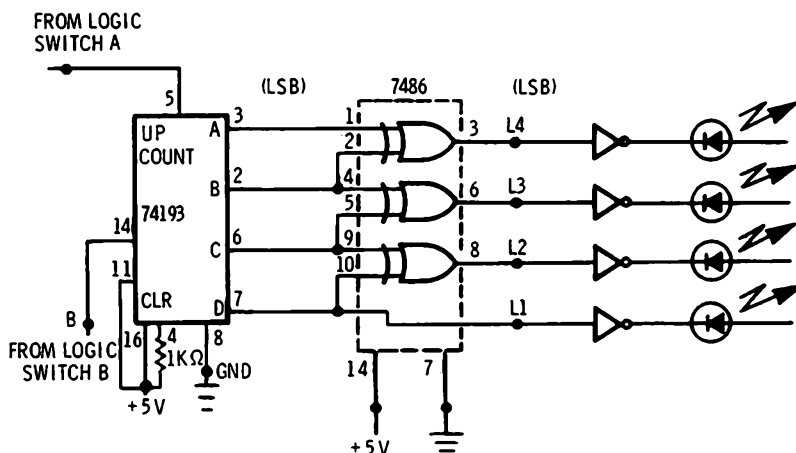


Figure 8-83
Code converter circuit
for Steps 6 through 8.

7. Apply power to the circuit. Depress the B logic switch momentarily to clear the counter to zero. Record the output code in the first space provided in Table II. Increment the counter sixteen times and at each step record the output code in Table II.
8. Observe the code that you copied in Table II. What code did the code converter circuit generate? _____

TABLE II

[illegible]

9. Wire the circuit shown in Figure 8-84. The 74LS86 quad X-OR circuit will be used as a code converter. The input code will come from data switches SW1 through SW4. You will observe the output code on LED indicators L1 through L4.

TABLE III

INPUTS				OUTPUTS			
SW1	SW2	SW3	SW4	L1	L2	L3	L4
0	0	0	0				
0	0	0	1				
0	0	1	1				
0	0	1	0				
0	1	1	0				
0	1	1	1				
0	1	0	1				
0	1	0	0				
1	1	0	0				
1	1	0	1				
1	1	1	1				
1	1	1	0				
1	0	1	0				
1	0	1	1				
1	0	0	1				
1	0	0	0				

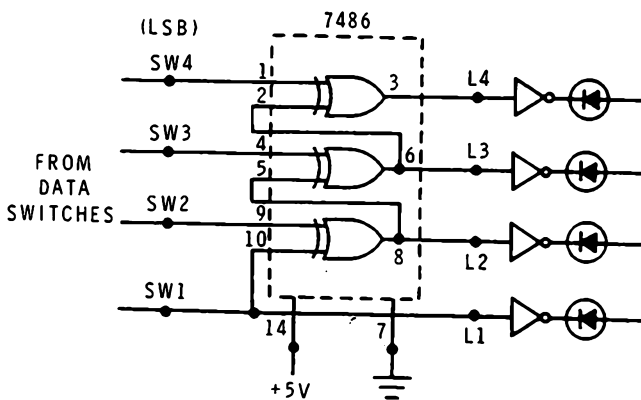


Figure 8-84
Code converter circuit
for Steps 9 through 11.

10. Apply the input code given in Table III to the code converter circuit. For each input code, record the corresponding output code in the spaces provided.
11. Study the input and output code in Table III. The input code is the _____ code.
The output code is the _____ code.

Discussion For Steps 6 Through 11.

In these steps you used exclusive OR gates to construct code converter circuits. The circuit in Figure 8-83 is a binary to Gray code converter. The binary code generated by the 74LS193 binary counter is applied to the 74LS86 exclusive OR circuit which generates the Gray code output. The code you recorded in Table II therefore is the Gray code. Check your results by referring back to the Gray code Table in Figure 8-51.

The circuit in Figure 8-84 is a Gray to binary code converter. You applied the Gray code to the input circuit with the data switches. The circuit should have generated the corresponding binary output code. Check your results by referring to Figure 8-51.

Procedure (Continued)

12. Construct the circuit shown in Figure 8-85. This is a four bit comparator circuit that monitors two four bit input words and generates an output signal that indicates when the two words are equal. The two four bit data sources are the data switches SW1 through SW4 and the four bit binary output code from the 74LS193 counter. The binary counter output is monitored on LED indicators L1 through L4. The comparator circuit is made up of the 74LS86, 74LS04, and 74LS20 ICs. The output is monitored on a DC voltmeter or logic probe.

Study the circuit in Figure 8-85 and determine the need for the 74LS04 inverters at the output of the exclusive OR gates. What are the purposes of these inverters? _____

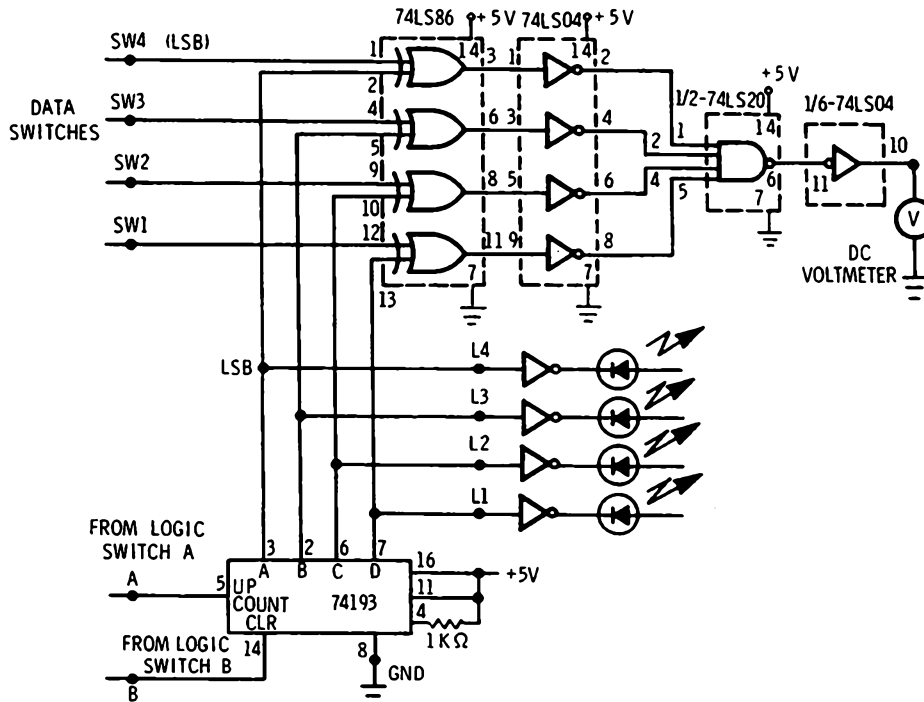


Figure 8-85
Comparator circuit
for Steps 12 and 13.

13. Apply power to the circuit. Depress the logic switch B momentarily to reset the counter to 0000. Verify the operation of the comparator circuit by setting the number 0011 into the data switches (SW4 = LSB). Then using the A logic switch, increment the 74LS193 counter. Observe the DC voltage at the output. As soon as the output voltage rises to the binary 1 level, compare the LED indicator display with the data switch state. They should be equal.

Set different values of four bit binary numbers into the data switches and again increment the counter with the A logic switch noting when the DC voltmeter reads the binary 1 voltage level at the output. For each set of input words, compare the LED indicator states to the data switch value when the output goes high.

Discussion For Steps 12 And 13

In these steps you constructed a binary comparator circuit and verified its operation. You compared the four bit binary word from the data switches to binary number generated by the 74LS193 counter. When the two numbers were equal, the DC voltmeter output indicates a binary 1 level.

A binary comparator circuit is constructed by using an exclusive NOR gate for each pair of corresponding bits in the words to be compared. The exclusive NOR gates are constructed in this circuit by connecting inverters at the outputs of the exclusive OR gates in the 74LS86. The outputs from the 74LS04 inverters are then ANDed in the 74LS20 NAND gate. The 74LS04 inverter at the output of the 74LS20 makes the circuit produce the AND function. The output is a binary 1 when the two four bit input words are identical.

Procedure (Continued)

14. Construct the circuit shown in Figure 8-86. This is a complex circuit so be careful in putting it together. There are six integrated circuits involved and many wiring connections. Be sure to connect +5 volts and ground to each IC. Don't forget to connect pins 2 and 7 on IC-74LS76 to +5V.

This circuit is a serial binary adder. Two 4-bit serial binary numbers are fed into a full adder circuit. One of the numbers to be added (the addend) is stored in the 74LS95 4-bit shift register. The other 4-bit number (the augend) is stored in the data switches and converted to serial form by the 74LS151 multiplexer. The adder is made up of the 74LS86 exclusive OR and the 74LS00 quad 2-input NAND gate. The two numbers are added a bit at a time and the sum is shifted back into the 74LS95 and displayed on LED indicators L1 through L4. One of the JK flip-flops in the 74LS76 IC is used as a memory for the carry bit (C_0) generated by the adder. The A logic switch is used to generate the clock or shift pulses. The B logic switch is used to reset the circuit.

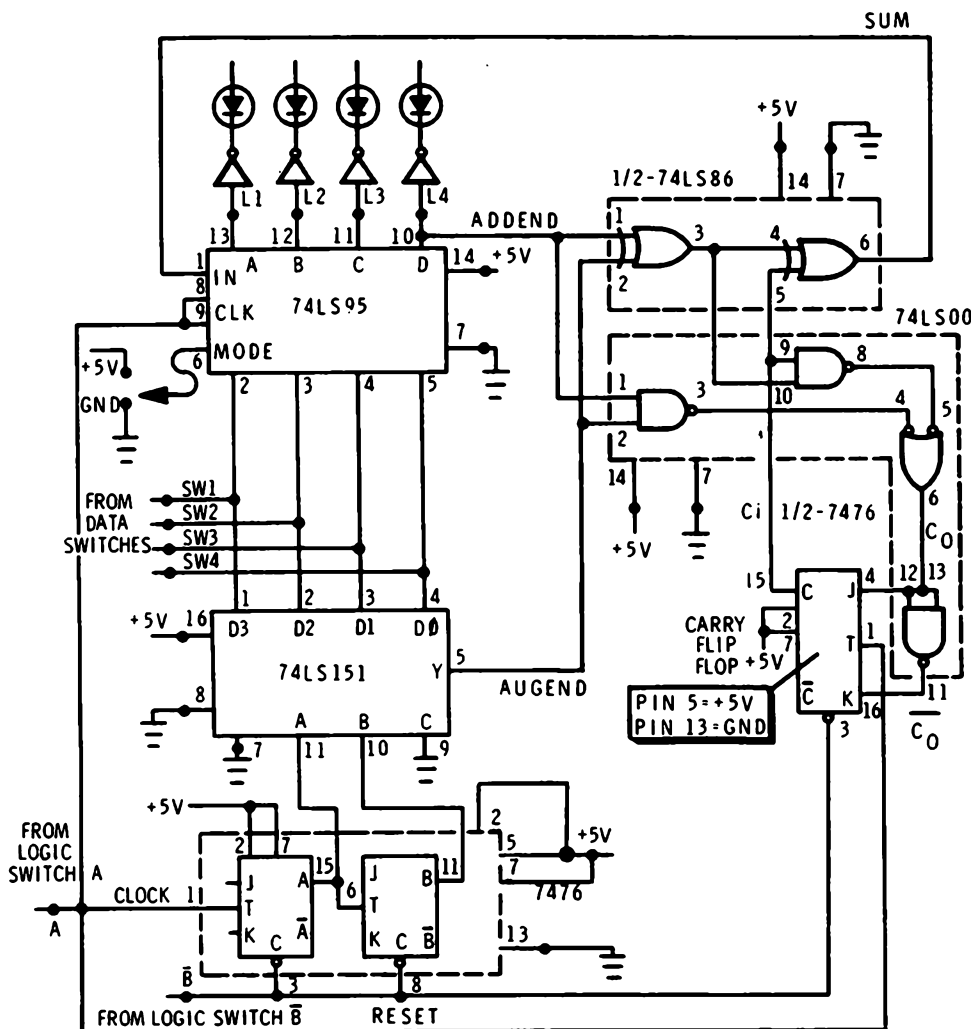


Figure 8-86
Serial binary adder
circuit for Steps 14 through 17.

15. Apply power to the circuit. Set the binary number 0111 into the data switches. Connect a wire from pin 6 of the 74LS95 shift register to +5 volts. This is the mode control input to the 74LS95 shift register that determines whether the device is parallel loaded or produces a shift right operation. In this step, you will load the register from the data switches. This will happen when the mode control input line is high. Depress the A logic switch once to load the data into the registers. The LED indicators should then read the same as the data switches, 0111.
Move the wire from pin 6 on the 74LS95 to ground. Set the data switches to 0101. Depress the B logic switch to reset the 2-bit binary counter driving the 74LS151 multiplexer and the carry flip-flop.
16. Depress the A logic switch four times. Each time you depress it, you will note a change in the LED display as the circuit generates the sum and shifts it into the 74LS95 register. Once four shift pulses have been generated, note the LED output display and record the number.
Sum = _____
17. Repeat step 15 several times, but use different binary numbers each time. Remember that, since the 74LS95 has a maximum content of four bits, the sum must be 15 (1111) or less if it is to appear correctly in the register. Repeat the operation several times to be sure that the circuit does perform properly.

Discussion For Steps 14 Through 17

The procedure for operating this adder circuit is to first load the 74LS95 shift register. You did this by setting the mode control at pin 6 to binary 1 to permit presetting of the register from the data switches. You depressed the A logic switch to generate the clock pulse that loads the 74LS95 register with the number 0111. You then returned the mode control to binary 0 so that the register would shift right.

Next, you set the number 0101 into the data switches. The 74LS151 multiplexer converts this parallel word into a serial word as it is sequenced by the two bit binary counter made up of a 74LS76 dual JK flip-flop IC.

Next, you used the A logic switch to generate shift pulses that cause the addition to occur. The number in the 74LS95 shift register is shifted out into the adder. These same pulses increment the binary counter and sequence the multiplexer. The sum is stored in the 74LS95. The correct sum should have been 1100.

The binary adder in this circuit operates just like the full adder circuit discussed previously in the unit. The only difference is the addition of a JK flip-flop from a 74LS76 IC to use as a carry memory. Since we are adding a single bit at a time, some means must be provided for remembering the occurrence of a carry so that it can be added to the next most significant bits in sequence.

The JK flip-flop stores this carry. Note that the output of one gate in the 74LS00 IC is labeled carry-out (Co). This is applied to the JK inputs. Assume that the LSBs of the numbers to be added are applied to the adder. The carry output line at this time will have on it a binary state that indicates the presence of a carry bit if the states of the input bit are such that it produces a carry. Assuming that a carry is produced, this carry signal will be loaded into the JK flip-flop when the first clock pulse occurs. At the same time, the adder is generating the proper binary sum of the two LSBs. This is applied to the serial input of the 74LS95 shift register. Therefore, when the first clock pulse occurs, the first sum bit is loaded into the 74LS95 shift register and the carry state is stored in the JK flip-flop.

The least significant bits have now been added and lost. The next most significant bits now appear at the adder inputs. The proper sum is generated and appears at the serial input of the 74LS95 shift register. At the same time, the carry state from the previous two bits stored in the JK flip-flop is applied to the carry input (Ci) of the adder. This ensures that the previously generated carry is added to the next two most significant bits. The proper sum then is generated and is loaded into the shift register when the next clock pulse occurs. Also during the next clock pulse time the carry state of the presently monitored bits is stored in the JK carry flip-flop. This action is continued until all four bits have been added. At this time the proper sum is in the 74LS95 shift register.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and answer all of the questions first before checking the answers.

1. Draw a logic gate decoder that recognizes the number 23.

2. A decoder has inputs $\overline{A}BC\overline{D}\overline{E}F$ where A is the LSB of a number. The number being recognized is:
 - a. 011001
 - b. 011000
 - c. 100110
 - d. 110010

3. A decoder with four inputs can have a maximum of how many outputs?
 - a. 4
 - b. 8
 - c. 16
 - d. 32

4. Which of the following cannot be used as a decoder?
 - a. inverter
 - b. NAND
 - c. AND
 - d. NOR

5. In a multiple output decoder made with NAND gates, which of the following is true?
 - a. The selected output will be high.
 - b. The selected output will be low.
 - c. Only one output will be high.
 - d. Only one output will be low.

6. Another name for a data selector is:
- decoder
 - demultiplexer
 - encoder
 - multiplexer
7. Parallel-to-serial data conversion can be performed with which of the following?
- demultiplexer
 - multiplexer
 - shift register
 - decoder
8. The input selected by a multiplexer is determined by a multibit input called the:
- input code
 - output code
 - address
 - channel number
9. The multiplexer in Figure 8-25 has inputs D0, D2, D3, D5, and D7 grounded. Inputs D1, D4, and D6 are connected to +5 volts. The sum-of-products Boolean output is:
- $\overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C}$
 - $A\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C$
 - $A\overline{B}C + \overline{A}\overline{B}\overline{C} + A\overline{B}C$
 - $A\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}C$
10. A circuit that generates a specific output code in response to an input is called a(n):
- encoder
 - decoder
 - code converter
 - data selector
11. The Boolean equation for an exclusive OR is:
- $XY + \overline{X}\overline{Y}$
 - $\overline{X}Y + X\overline{Y}$
 - $XY + X\overline{Y}$
 - $\overline{X}Y + \overline{X}\overline{Y}$
12. A single bit comparator is called a(n):
- exclusive OR
 - wired OR
 - exclusive NOR
 - NOR gate

13. The circuit that detects bit errors in a binary word is called a:
- comparator
 - exclusive NOR
 - decoder
 - parity detector
14. The parity bit added to the word 101011010110 to create even parity is:
- 0
 - 1
15. Add the following binary numbers:
- | | | | |
|----|----------------|----|----------------|
| a. | 10010 | b. | 11111 |
| | <u>+ 11011</u> | | <u>+ 10001</u> |
16. The circuit that would change the pure binary code into ASCII is called a(n):
- code converter
 - decoder
 - encoder
 - demultiplexer
17. The input applied to a ROM is called the:
- input code
 - address
 - data
 - micro instruction
18. How many bytes of data can be stored in a 16384 bit ROM?
- 512
 - 1024
 - 2048
 - 4096
19. An LSI circuit that implements multiple sum-of-products Boolean equations is called a(n):
- multiplexer
 - ROM
 - decoder/demultiplexer
 - PLA
20. The main element of a microprogrammed controller is a(n):
- address register
 - PLA
 - ROM
 - binary counter

21. A ROM has six inputs and eight outputs. How many bits does the ROM store?
 - a. 512
 - b. 1536
 - c. 2048
 - d. 4096
22. Draw the diagram of a decoder to recognize the number ABCD = 1011. There is more than one possible solution.
23. Which of the following is **not** a good ROM application?
 - a. look-up table
 - b. code conversion
 - c. squaring (X^2)
 - d. addition
 - e. microcontroller
24. A combinational logic circuit whose multiple sum-of-product outputs can be programmed by the user is a(n):
 - a. multiplexer
 - b. decoder
 - c. PLA
 - d. ROM
25. Which of the following is **not** an application for an exclusive OR?
 - a. code generation
 - b. complementing
 - c. addition
 - d. parity generation
 - e. code conversion

EXAMINATION ANSWERS

1. See Figure 8-87

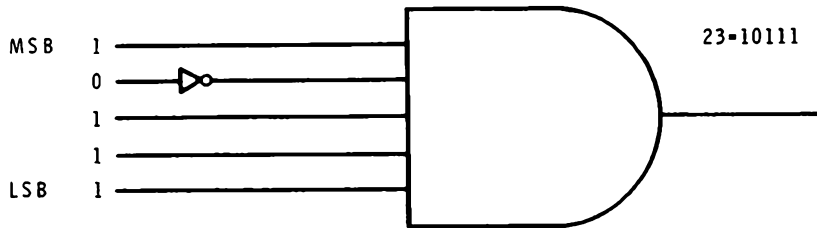


Figure 8-87
Solution to Question 1.

2. c.-100110
 3. c.-16 $2^4 = 16$
 4. a.-inverter
 5. b.-The selected output will be low.
 d.-Only one output will be low.
 6. d.-multiplexer
 7. b.-multiplexer
 c.-shift register
 8. c.-address
 9. d.- $\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC$
 10. a.-encoder
 11. b.- $\overline{X}Y + X\overline{Y}$
 12. c.-exclusive NOR
 13. d.-parity detector
 14. b.-1
 15. a. $\begin{array}{r} 10010 \quad 18 \\ + 11011 \quad 27 \\ \hline 101101 \quad 45 \end{array}$
 b. $\begin{array}{r} 11111 \quad 31 \\ + 10001 \quad 17 \\ \hline 110000 \quad 48 \end{array}$
 16. a.-code converter
 17. b.-address
 18. c.-2048 One byte = 8 bits
 $16384 \div 8 = 2048$
 19. d.-PLA
 20. c.-ROM
 21. a.-512 $2^6 = 64, 64 \times 8 = 512$

22. See Figure 8-88

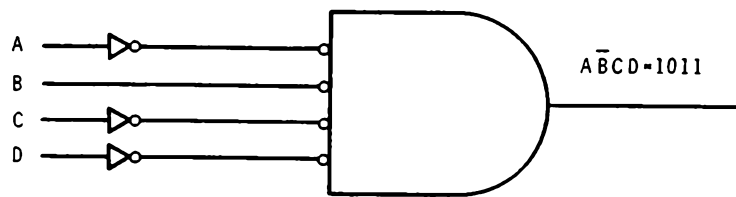


Figure 8-88
Solution to Question 22.

23. d.—addition

24. c.—PLA

25. a.—code generation

Unit 9

**SEMICONDUCTOR
MEMORIES**

CONTENTS

Introduction	9-3
Unit Objectives	9-4
Unit Activity Guide	9-5
Memory Types and Organization	9-6
Memory Characteristics and Specifications	9-17
Read/Write Memories	9-25
Dynamic Memories	9-45
Programmable Read-Only Memories	9-61
Experiment 23 — Semiconductor Memories	9-73
Unit Examination	9-81
Examination Answers	9-88

INTRODUCTION

Semiconductor memories are integrated circuits that store digital data. Like flip-flop registers, semiconductor memories can store multi-bit binary words. However, semiconductor memories are capable of storing many thousands of these binary words. They are used primarily in digital computers to store programs and data. You will also find integrated circuit memories in almost all electronic equipment implemented with digital circuits. For this reason, no study of digital techniques is complete without coverage of semiconductor memories.

There are two basic types of semiconductor memories: RAMs and ROMs. RAMs (random access memories) are devices where data can be stored and later retrieved. Also called read/write memories, RAMs are used for the temporary storage of binary data. Typical RAMs can store from several hundred bits to hundreds of thousands of multi-bit binary words. All of these words are randomly accessible in less than a microsecond.

ROMs (read only memories) are devices where binary data is permanently stored and can be accessed whenever it is needed. Therefore, the data stored in ROM cannot be lost or replaced by other data. You have already learned about ROMs in Unit 8. Additional coverage of ROMs is included in this Unit. You will learn the various types of RAMs and ROMs, how they work, and how they are used. Typical integrated circuit memories will be introduced and discussed.

UNIT OBJECTIVES

When you complete this Unit, you will be able to:

1. Draw a block diagram of the hierarchy of semiconductor memories.
2. Describe the organization of random access read/write and read only memories.
3. Name the two major types of read/write memories, describe how they operate, and compare and contrast their characteristics and applications.
4. Name the types and explain the operation of programmable read only memories.

UNIT ACTIVITY GUIDE

	Completion Time
<input type="checkbox"/> Read "Memory Types and Organization".	_____
<input type="checkbox"/> Answer Self Test Review questions 1-15.	_____
<input type="checkbox"/> Read "Memory Characteristics and Specifications".	_____
<input type="checkbox"/> Answer Self Test Review questions 16-21.	_____
<input type="checkbox"/> Read "Read/Write Memories".	_____
<input type="checkbox"/> Answer Self Test Review questions 22-34.	_____
<input type="checkbox"/> Read "Dynamic Memories".	_____
<input type="checkbox"/> Answer Self Test Review questions 35-46.	_____
<input type="checkbox"/> Read "Programmable Read-Only Memories".	_____
<input type="checkbox"/> Answer Self Test Review questions 47-62.	_____
<input type="checkbox"/> Perform Experiment 23.	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Review the Examination Answers.	_____

MEMORY TYPES AND ORGANIZATION

Memories are electronic circuits that store large quantities of binary data. Over the years, a wide variety of electronic circuits have been used to implement digital memories. Delay lines, cathode ray tubes, and magnetic drums were common in the very early digital computers. Relays, vacuum tubes, and magnetic tape have also been used as storage media for binary data. For many years, the most common memory element was a tiny doughnut shaped magnetic core. This core can be magnetized in one of the two directions to store a single binary bit. Many thousands of these tiny cores were organized and interconnected to form memories where thousands of multi-bit binary words could be stored. Magnetic core memories are still found in many of the older computers in use today. And, they are, in fact, still used in some critical, high-reliability computers for military and space applications.

For the most part, all types of electronic memories have been replaced by semiconductor memories in new computers and other digital equipment. High density integrated circuit memories were originally developed in the early 1970's, and during the past decade have been refined to a considerable degree. Today, semiconductor memories capable of storing hundreds of thousands of bits of data on a single tiny chip are available for extremely low prices. Integrated circuit memory chips have become the most reliable and economical form of binary storage, regardless of the application. Because semiconductor memories predominate today, this Unit focuses only on the various types of semiconductor memories.

Memory Classification

Figure 9-1 shows the hierarchy of semiconductor memories. The two basic types of semiconductor memories are read/write memories and “read only” memories. Both types are made up of many individual storage elements, each capable of storing one bit of data. Flip-flops and integrated capacitors are typically the storage elements used.

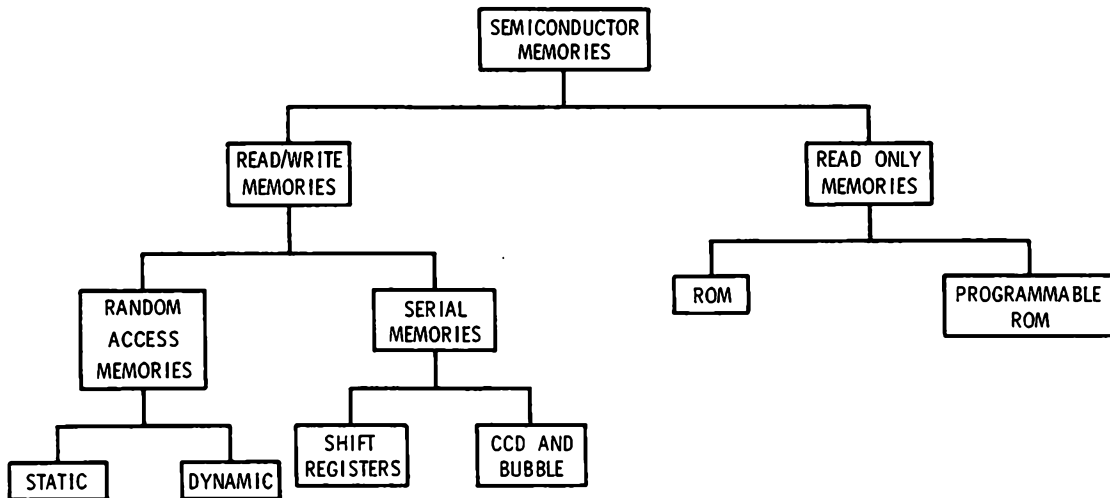


Figure 9-1

Hierarchy of Semiconductor memories.

Read/write memories are devices into which data can be stored and then later retrieved. When data is stored in a memory, it is said to be written into it. When the data is to be retrieved from the memory, it is said to be accessed or read. Read/write memories are simply temporary storage locations for binary words. Binary data is stored in the memory for some undetermined length of time, depending upon the application. Later, if that data is needed, it can be accessed and read out of the memory into other digital circuitry for processing.

The binary data normally stored in a read/write memory can take on many different meanings. Data words stored in the memory may simply be ASCII coded text, numerical data, or computer instructions that form a program. The memory itself pays no attention to the exact meaning of the data. It is up to the external circuitry connected to the memory to interpret the data and process it properly.

As you can see in Figure 9-1, there are two primary classifications of read/write memories, random access memories (RAMs) and serial memories. RAMs store parallel binary data that can be accessed at random. Any one of thousands of binary words can be located directly without reference or interference to all other binary words. The two types of RAMs are static and dynamic. You will learn about both types later in this Unit.

Serial memories store serial binary data. To access any desired binary word, you must sequence through all other words until you find it. The classifications of serial memories are shift registers, special devices known as charge coupled devices (CCDs), and magnetic bubble memories. Shift register memories will be covered later, but CCDs and bubble memories will not be discussed as they are largely experimental and not widely used.

Of the two types of memories, RAMs are faster and the most desirable, but they are also more expensive. Serial memories are slower but somewhat less expensive. Because RAMs are by far the most widely used, we will emphasize static and dynamic RAMs in this Unit.

Refer again to Figure 9-1. Another major class of memory is the read only memory, or ROM. As its name implies, a ROM can perform the read function only. Data is permanently stored in a ROM usually when it is manufactured. In other words, the desired data words, computer instructions, and other binary information is initially written into the memory and cannot be changed. However, information can be accessed and read out at any time.

ROMs are far more economical, faster, and can store greater volumes of data than read/write memories. They are very commonly used in digital equipment to store frequently used binary information. As with a read/write or random access memory, the ROM can store numbers, text, computer instructions, or useful bit patterns of some kind. In addition, you have also seen in a previous Unit how ROMs can be used to perform a variety of useful digital logic functions. In any case, the data in a ROM cannot be changed.

Read only memories have already been discussed in Unit 8. Later in this Unit we will expand upon that coverage, introduce programmable read only memories, and show how this special class of ROM can be programmed in the field.

Memory Organization

There are two major ways in which semiconductor memories are organized and data accessed. The first is the serial memory with sequential access to data, and the second is the parallel or random access method. Both types are widely used in digital equipment, but the parallel or random access memories predominate because of their higher speed and simplicity. Serial memories are used in applications where slower speeds and lower cost are important.

SEQUENTIAL ACCESS

Serial memories are circuits in which data is stored and retrieved sequentially. The earliest successful semiconductor memories were sequential access devices. A good example of a typical semiconductor sequential access memory is a shift register. Large shift registers contain many flip-flops or capacitive storage elements and can store many bits of data. For example, a typical early shift register memory was capable of storing 1024 bits of data. A 1024-bit shift register can, for example, store 128 bytes of data ($1024/8 = 128$). Recall that a byte is an 8-bit word. Shift registers containing from several hundred to many thousands of storage elements are widely available.

Figure 9-2 shows the organization of a typical sequential access shift register memory. Assume a total shift register length of 1024 bits. Further assume that we will store 128 bytes of data. These 128 bytes, numbered 0 through 127, are stored end to end in adjacent shift register storage elements. To store data, the desired byte is shifted into the shift register a bit at a time. To access a byte of data, the desired word is shifted out a bit at a time.

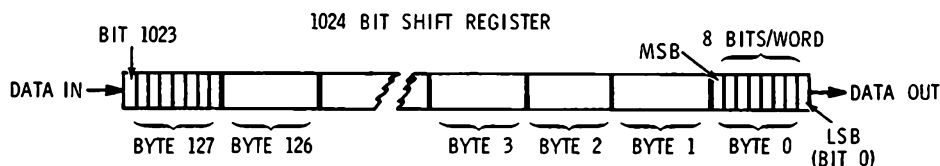


Figure 9-2
Storing data words sequentially in a serial
shift register memory.

The 1024 storage elements are divided into 128 locations where 8-bit words can be stored. Each word location is given a number, called the address, between 0 and 127. Address 0 is the first location in memory while address 127 designates the last storage location in the memory. Data is entered and retrieved with the LSB (least significant bit) of the data word being stored and accessed first. By numbering the various memory locations, you can easily access a desired piece of data which you may have stored in a specific location.

The circuitry used for controlling read/write operations and locating a selected binary word is shown in Figure 9-3. The bit and address counters keep track of the memory contents on a bit and word basis. The bit counter is a three-bit binary counter whose input is connected to the clock signal that operates the shift register. Recall that data is shifted to the right one bit at a time for each clock pulse. Recall also that a three bit counter has eight possible states (000 through 111), one state for each bit in a memory word. For every eight clock pulses, the bit counter recycles, indicating that another 8-bit word has been shifted eight positions to the right.

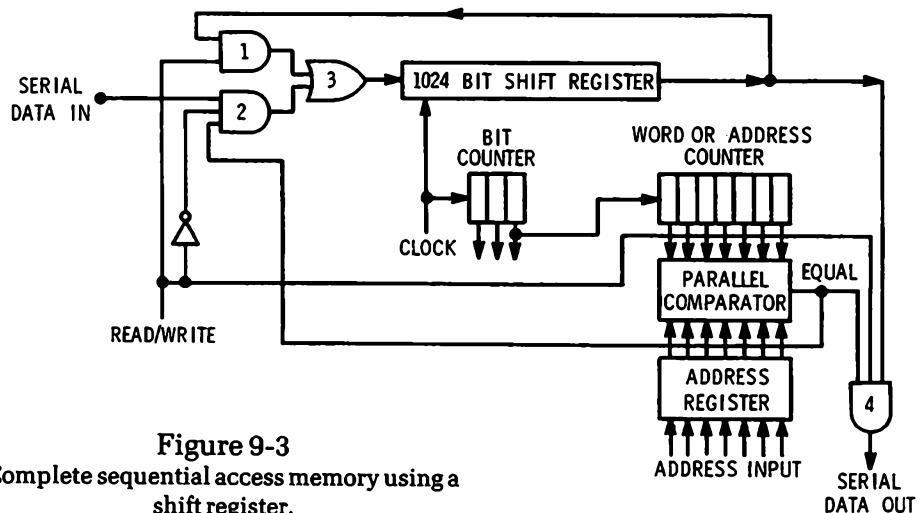


Figure 9-3
Complete sequential access memory using a shift register.

The bit counter drives the 7-bit word or address counter. Each time the bit counter recycles, it toggles the address counter which keeps track of the number of words shifted. The output of this 7-bit counter is a number that designates the location of a specific place in the memory where data is stored.

Assume that both the bit and word counters are initially reset. The address, therefore, is 0000000. This is the binary address for memory location 0 in the memory. At this time, the LSB of byte 0 appears at the output of the shift register.

Now assume that clock pulses are applied to the shift register. As each clock pulse occurs, the bits of the byte stored in location 0 are shifted out of the register a bit at a time. After 8 clock pulses have occurred, byte 0 will have been completely shifted out of the register. On the 8th clock pulse, the bit counter recycles from 111 to 000 and, therefore, toggles the address counter so that its content is now 0000001. This is the address for the next word in sequence at location 1. At this time, the LSB of byte 1 appears at the output of the shift register.

Now that you have some idea as to the organization and operation of a shift register sequential memory, let's take a more detailed look at both read and write operations.

Refer again to Figure 9-3. Gates 1, 2, and 3 form a two input multiplexer. The input to gate 2 of the multiplexer is the external serial data that is to be stored in memory. The other input to the multiplexer is applied to AND gate 1. This input is derived from the output of the shift register. During a read operation, bits are shifted out serially, usually into another shift register for storage and then further processing. In order that the word not be lost during the read process, the data is simply shifted back into the shift register. This is known as data recirculation. The 128 bytes of data stored in the shift register are continuously recirculated so that none of it will be lost.

The remaining circuitry in Figure 9-3 consists of an address register and a comparator. To identify a specific memory location, a parallel binary address is loaded into the address register. The output of this register is applied to a 7-bit comparator. The other input to the comparator is the output of the 7-bit address or word counter. To locate the binary word stored at location 5, for example, the number 0000101 is loaded into the address register. This number in turn appears at one of a set of inputs to the comparator. Clock pulses are applied to the shift register and the bit counter. The address counter is incremented until it contains the number 0000101. At this time, the comparator output EQUAL is binary 1 indicating a match. This output is used as one of the enable inputs to output AND gate 4. If a read operation is taking place, the read/write input line will be high, furnishing an additional input enable to AND gate 4.

To perform a read operation, the read/write control line is set high. This enables gates 1 and 4. The bits of the word stored at location 5 are shifted out of the register serially and passed through gate 4 to the output. The data is also recirculated through gate 1 so that it will not be lost.

A write operation is similar, since the location into which data is stored is entered into the address register. The clock pulses occur and increment both the bit and address counters. When the comparator signals a match, it enables AND gate 2. The read/write control line is binary 0 at this time, thereby enabling gate 2 through the inverter. The word to be stored is applied to gate 2. As clock pulses occur, the external word is shifted in a bit at a time, LSB first. After 8 clock pulses occur, the address counter is incremented. The comparator output goes to binary 0, inhibiting gate 2 and ending the write operation.

While shift register memories are not as widely used as they once were, you will still find them in some types of digital equipment. They are particularly useful where the data is already in serial form. Sequential access memories are slow because you do not have direct access to a given word. You must step through all of the bits in the various words until you come to the one that you want. This greatly slows down both read and write operations. However, for some applications speed is of less importance and sequential access is satisfactory. With today's high speed shift registers which can operate at speeds in excess of 20 MHz, even sequential access is relatively fast.

In some instances, you will also find that sequential access memories are also lower in cost. In those unique applications where serial data, low cost, and moderate access speeds are required, sequential access memories will suffice.

RANDOM ACCESS

Random access memories, like sequential access memories, are organized as many storage locations for fixed length binary words. However, in the random access memory each word is directly accessible. Any word location is randomly addressable and data may be either written into, or read from, that location. It is not necessary to sequence through all other words in memory to get to the one you want. Reading or writing operations usually take place with parallel rather than serial data. Because of this form of organization, memory speed is much higher than that of sequential access memories. Any given memory location can be addressed and data can be stored into that location or read from it in 500 nanoseconds or less.

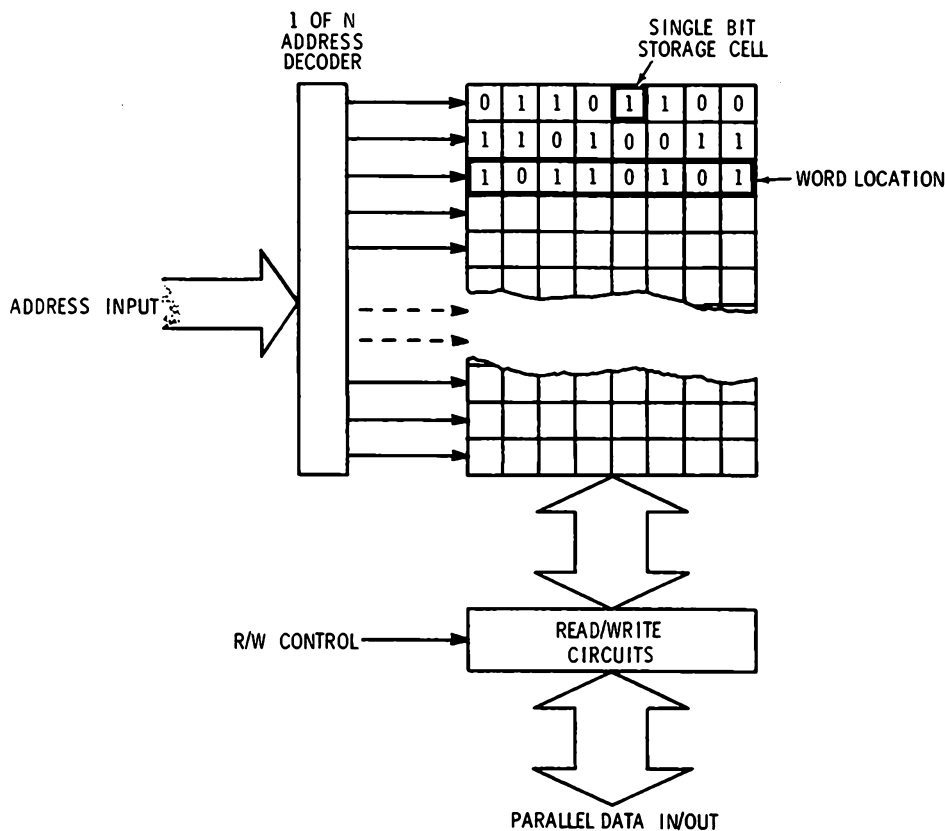


Figure 9-4
Random access memory organization.

Figure 9-4 is a simplified block diagram of a random access memory. The memory is made up of storage cells grouped to form locations capable of storing a fixed length binary word. In most digital and microcomputer systems, an 8-bit word length is typical. Most memory ICs contain thousands of such storage locations. Each storage location is similar to an 8-bit register. While some semiconductor memories do use flip-flops as the basic storage cell, other types of semiconductor memories do not. More on this later.

Each of the storage locations is given a number called the address. The address is a parallel binary number that designates a specific storage location. The parallel address word is applied to a decoder circuit that enables one of the memory locations.

To use a random access memory, an address word is applied to the device. A read/write (R/W) control line on the memory selects the operation to be performed. If a read operation is selected, data from the memory location enabled by the decoder is read out in parallel. Usually this data is loaded into an external register for further processing. If a write operation is selected, the data to be stored is applied to the parallel data input lines. The word is then stored in the memory location enabled by the decoder.

It is important to point out that the basic random access memory organization shown in Figure 9-4 applies to ROMs as well as to read/write memories. Read/write memories are called RAMs even though ROMs are also random access in nature.

Self Test Review

1. The two basic types of semiconductor memories are:
 - A. _____
 - B. _____
2. The general name for the device that stores one bit of data is:
 - A. register
 - B. word location
 - C. storage cell
 - D. flip-flop
3. To store data in a RAM, a _____ operation is performed.
4. To retrieve data from a RAM, a _____ operation is performed.

5. Memories that can store and retrieve data are called _____.
6. Memories into which data is permanently stored are called _____.
7. Memories that store and retrieve data serially are referred to as _____ memories.
8. The main storage device in a serial memory is a _____.
9. The number of BCD digits that can be stored in a 2048-bit serial memory is _____.
10. To prevent the loss of serial data during a read operation, the data is _____.
11. The binary number used to designate the location of a specific word in memory is called the _____.
12. Any data word can be retrieved directly in a _____ memory.
13. The binary format of the input/output data used with a RAM is _____.
14. The circuit in a RAM that selects the desired word is called the _____.
15. A ROM is also a _____ memory.

Answers

1. A — Read/Write (RAM)
B — Read only (ROM)
2. C — storage cell
3. write
4. read
5. read/write or random access memories
6. read only memories
7. sequential access
8. shift register
9. 512 $2048 \div 4 = 512$ Each BCD digit is 4 bits.
10. recirculated
11. address
12. random access
13. parallel
14. decoder
15. random access

MEMORY CHARACTERISTICS AND SPECIFICATIONS

All semiconductor memories have certain characteristics and specifications that indicate their performance and capabilities. These include volatility, access time, size, and organization. Let's consider each of these characteristics in more detail.

Volatility

Volatility is a common characteristic of most read/write semiconductor memories. Volatility refers to the loss of all stored data when power is removed. Keeping in mind that the storage cells for binary data in semiconductor memories are electronic circuits or components, it is understandable that when power is not applied to them, they cannot function. For example, as long as power is applied to a flip-flop it can be set or reset to store one bit of data. A capacitor can be charged or discharged to represent a binary 1 or 0. However, when no power is available, charge and discharge operations cannot be performed. As long as power supplies furnish the rated voltage, semiconductor memories perform superbly. But even a minor power line aberration or a momentary power loss will cause data to be lost. If the power is interrupted for even microseconds, most semiconductor RAMs will be erased.

Volatility is a characteristic that is undesirable in semiconductor memories. However, little can be done to overcome it. In most applications this is not a great disadvantage. The standard AC power line is incredibly reliable and only in rare cases can power failure cause a disruption of service and the loss of valuable data. In computers and digital equipment using semiconductor memories, power supplies are designed with extra filtering and regulation capabilities that help eliminate minor short-duration power glitches. In critical applications, back-up power supplies are provided. In some equipment, for example, battery supplies are provided. These cut in automatically should a power failure occur, and cause the data to be retained until power is restored. Large computer systems often have separate motor-generator sets for maintaining continuous power independent of commercial power lines.

Read only memories are non-volatile. Because data is permanently stored in a ROM when it is manufactured, the loss of power does not destroy the memory contents. This characteristic also applies to programmable read only memories. These are memories which can be programmed after they are manufactured. In critical applications where data must be retained, a ROM is often used.

Access Time

Access time is the time interval between the instant at which data is requested from memory and the instant at which the information is available. Access time, also referred to as read time, is that short period between the time the desired word in a semiconductor memory is addressed and the time that word appears at the memory output. Typically a word is requested by applying a binary address to the memory device. The binary address plus other control signals are the inputs to the memory device that signal the request for information. Keeping in mind that a semiconductor memory is nothing more than a digital logic circuit, you can see that it requires a finite period of time for these logic signals to propagate through the various gates and circuits in the memory. Some time later, the requested information becomes available at the output.

The access time of most semiconductor memories is measured in nanoseconds. In some very high speed bipolar memories, access time is less than 20 nanoseconds. In conventional MOS LSI memory devices, access times from 100 to 500 nanoseconds are typical.

The access time is sometimes defined in terms of the time interval required to store data in a random access read/write memory. The write time is that interval between the time data is applied to the memory for storage and the instant at which the storage function is complete. Again, to store data in a memory, the address of the desired storage location is applied to the memory device. The data to be stored is also applied. After the various propagation delays have occurred, the actual storage function takes place. Write times are also very short and range from a few nanoseconds to several hundred nanoseconds depending upon the type of memory device used.

Memory Size

Memory size refers to the number of bits of data that a semiconductor memory can store. The earliest form of semiconductor memories could store only a few bits of information. They weren't much larger than a shift register. A typical early TTL semiconductor memory had a maximum storage capability of 64 bits, for example. Today, memory ICs capable of storing up to 1,000,000 bits are commonly available. As semiconductor technology improves year after year, storage capacities increase regularly. Most semiconductor memory ICs store a number of bits that is some power of 2. Since the address word applied to a semiconductor memory is a binary number, it is convenient and desirable to have the address word define the total number of memory locations. For example, an 8-bit address word can define 256 memory locations. A 12-bit address can define 4096 memory locations or bits of data.

Table I shows the relationship between number of address bits and the total number of bits of storage that can be addressed. The third column is a shorthand notation for designating memory size. Note that a 4096-bit memory is also referred to as a 4K memory. K in electronics shorthand generally means 1,000. A 4K memory then would be one with 4,000 bits of storage. However, because we are using powers of 2, the designation K when used with respect to memories refers to 1024. Therefore, a 4K memory means 4096 bits of storage.

TABLE I		
NUMBER OF ADDRESS BITS	NUMBER OF BITS OF STORAGE	NUMBER OF BITS (SHORTHAND)
8	256	256
9	512	512
10	1,024	1K
11	2,048	2K
12	4,096	4K
13	8,192	8K
14	16,384	16K
15	32,768	32K
16	65,536	64K

K = 1024

Memory Configuration

Memory configuration refers to the ways that the bits of storage in the semiconductor memory device are organized into storage locations for multi-bit words. The size of a memory IC tells only the total number of available bits of storage. The size figure does not indicate how those bits are arranged. For example, consider a semiconductor memory capable of storing 1024 bits. This 1K memory could be organized as 1024 memory locations for 1-bit binary words. This organization is generally indicated by a shorthand designation 1K by 1 or $1K \times 1$.

Another arrangement of the 1K memory could be 256 four bit words. Again the total number of bits is 1024, but the memory is simply organized to store 256 parallel 4-bit words. This designation is 256×4 .

There are many different kinds of memory configurations. The larger the number of bits, the greater the number of possible variations. However, only certain configurations are popular and practical. For example, in a 16K bit memory, a $16K \times 1$ organization is popular. Memory chips with this organization can then be grouped together to form larger memories. For example, one $16K \times 1$ device can be used for each bit in a multi-bit memory. If eight $16K \times 1$ chips are used, a memory capable of storing 16K bytes is created. This would be designated as a $16K \times 8$ memory.

Table II shows the various sizes of semiconductor memory chips commonly available and some of their more popular organizations.

TABLE II		
NUMBER OF BITS PER CHIP	TYPICAL RAM CHIP CONFIGURATIONS	TYPICAL ROM CHIP CONFIGURATIONS
1K	1K × 1 256 × 4 128 × 8	256 × 4
2K	—	512 × 4
4K	4K × 1 1K × 4	1K × 4 512 × 8
8K	1K × 8	1K × 8
16K	16K × 1 4K × 4	2K × 8
32K	—	4K × 8
64K	64K × 1	8K × 8
128K	—	16K × 8
256K	256K × 1 64K × 4 32K × 8	32K × 8
1M	1M × 1 256K × 4	128K × 8 64K × 16

Device Technology

Another characteristic of IC memories to consider is the semiconductor technology used in making them. The two basic categories of device technology are bipolar and MOS.

Bipolar RAMs and ROMs are available in both TTL and ECL versions. Bipolar RAMs and ROMs are not widely used because of their limited storage capability, high cost, and high power consumption. However, they are used where high speed is needed. ECL RAMs have access times as low as 5 nanoseconds. TTL RAMs and ROMs have access times in the 10 to 50 nanosecond range.

Most semiconductor memories use MOSFET technology. The small size of a storage cell permits thousands of bits of data to be contained on a chip. MOS circuits also have lower power consumption than bipolar devices and cost less per bit than any other type. N-channel MOS devices are predominant.

CMOS RAMs are also available. Their main benefit over other types is extremely low power consumption. CMOS RAMs are ideal for portable, battery-operated digital equipment.

Self Test Review

16. A memory is said to be _____ if it loses data when power is removed.
17. The time it takes to address and read out a word in memory is called _____ time.
18. A 14-bit address can specify how many storage locations?
 - A. 4K
 - B. 8K
 - C. 14K
 - D. 16K
19. The word size of a $1K \times 4$ memory is _____ bits.
20. The total storage capacity of a $4K \times 8$ ROM is _____ bits.
21. Most semiconductor memories are made with which kind of devices?
 - A. bipolar
 - B. MOS

Answers

16. volatile

17. access

18. D — 16K or 16,384

19. 4

20. 32K $4K \times 8 = 32K$

21. B — MOS

READ/WRITE MEMORIES

Read/write memories are most often referred to as random access memories or RAMs. In this Unit, we will use the designation RAM to refer to read/write memories. Since both read/write and read only memories have random access organizations and operation, it is unfortunate that the term RAM has been designated for read/write memories. Nevertheless it is the standard reference and, therefore is used here.

There are two basic types of semiconductor RAMs: static and dynamic. All semiconductor memories are made up of individual memory cells, each capable of storing a single bit of data. A variety of different types of components and circuits are used to implement these memory cells. There are both static and dynamic cells. The main storage element in static RAMs is a flip-flop or latch. In a dynamic cell, the storage element is a capacitor. This is either the small distributed capacitance between the gate and the source of a MOSFET or a tiny separate integrated capacitor. The capacitance is several picofarads or less. But with the very high impedances of MOSFET circuitry, it is capable of holding a stored charge long enough to make it a reliable storage media.

In a static RAM, the gate-to-source capacitances are used to control MOSFETs that are connected as a latch. The latch or set/reset flip-flop can store a single bit of data. The latch is either set or reset by external signals.

In a dynamic RAM, a capacitor is the main storage element. This small capacitor operates a MOSFET whose output state is sensed to determine whether a binary 0 or a binary 1 is stored. When the capacitance is charged, a binary 1 is stored. When the gate capacitance is discharged, a binary 0 is stored. Because of leakage resistance across the capacitance, the charge can leak off and result in lost data. To prevent this, dynamic memory cell capacitors must be periodically recharged, or refreshed, in order to retain their state.

In this section, you will study typical static and dynamic semiconductor memory circuits.

Static Memories

The basic static RAM storage cell is a flip-flop. A typical circuit is shown in Figure 9-5. The storage cell is made up of six enhancement mode MOSFETs. The main storage circuit is a flip-flop consisting of cross-coupled inverters Q1 and Q3. Q2 and Q4 are MOSFETs connected as active load resistances. Note the source to gate capacitances of the inverters which store the basic charges that keep the flip-flop in either the set or reset state.

The two basic states of this flip-flop are when Q1 is conducting and Q3 is cut off and when Q3 is conducting and Q1 is cut off. The state of the MOSFETs is determined by the charge on the gate capacitances. If the gate capacitance of Q1 is charged to the supply voltage, V_{DD} , the conduction threshold of Q1 will be exceeded and Q1 will conduct. Current will flow through Q1 and its load device Q2 to the supply voltage. The very low resistance of Q1 causes its drain at point A in the circuit to be very near ground.

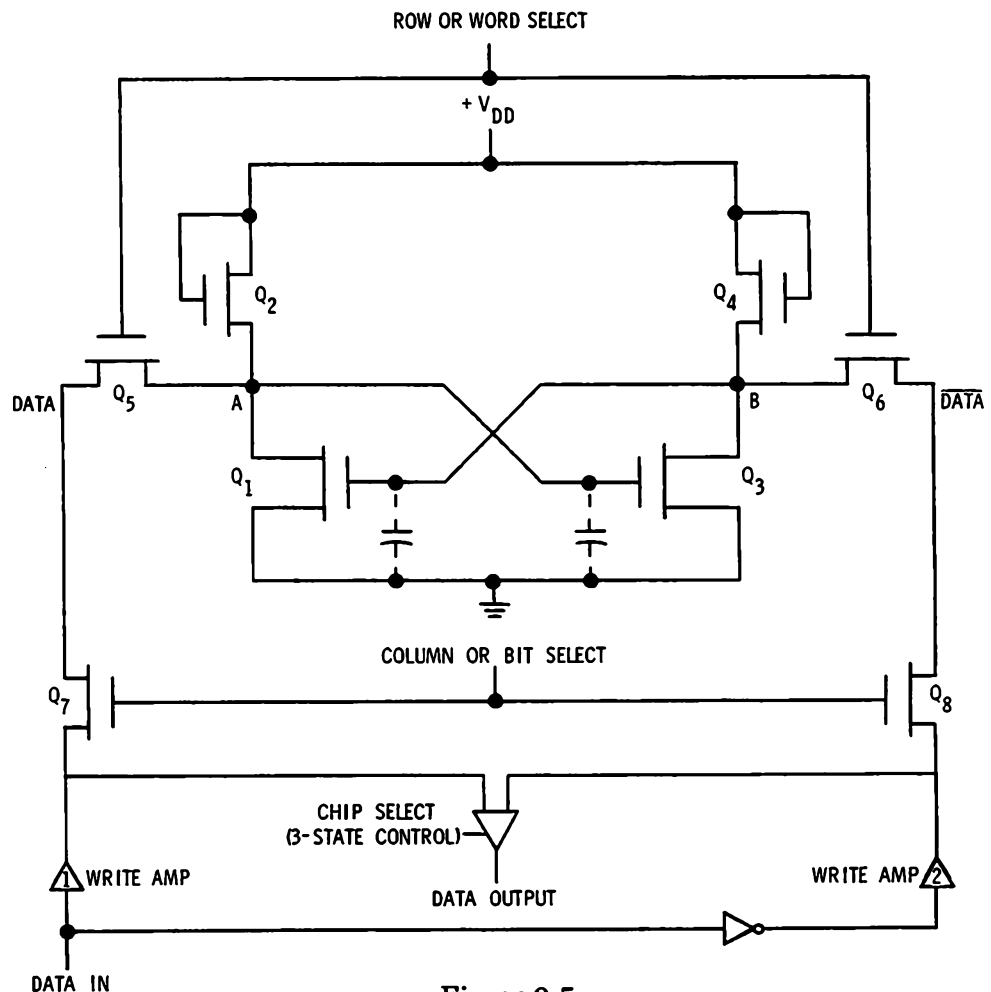


Figure 9-5
Typical static RAM storage cell.

Point A is also connected to the gate of Q3. Since the voltage at point A is below the conduction threshold of Q3, Q3 will be cut off. Its drain at point B will be approximately V_{DD} as seen through its conducting load device Q4. Since Q4 is conducting, it keeps the gate capacitance of Q1 charged and, therefore, Q1 continues to conduct. This stable state of the latch causes one binary state to be stored. The opposite binary state is achieved when Q3 conducts and Q1 is cut off.

Two common variations of the basic static storage cell are shown in Figure 9-6. In Figure 9-6A, load MOSFETs Q2 and Q4 are depletion mode rather than enhancement mode devices. Faster circuit operation is achieved with depletion mode loads. In Figure 9-6B, resistors replace the MOSFET loads. These are tiny polysilicon resistors. Their high value reduces power consumption and their small size helps increase bit density for a given chip size. All three cell types are widely used.

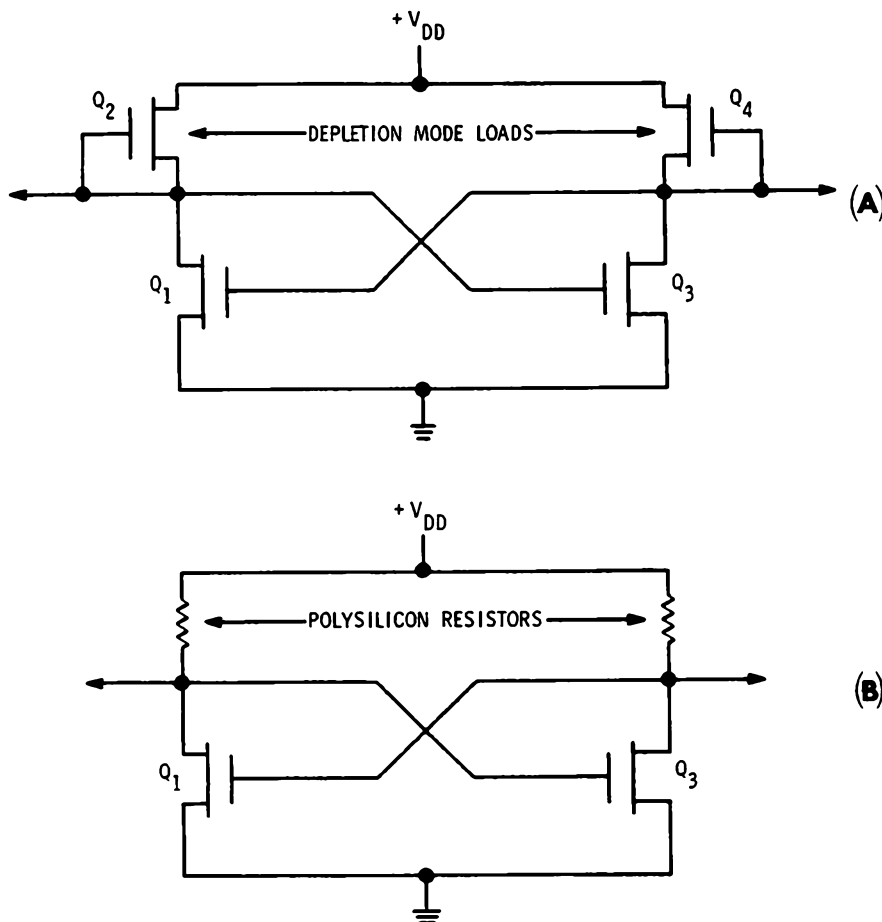


Figure 9-6
Variations in static RAM cells.
(A) Depletion mode load MOSFETs.
(B) Polysilicon resistor loads.

There are three basic conditions that occur in a semiconductor memory. The first condition is where the storage cell is simply retaining the information for future use. Another condition is where a read operation is being performed. It is at this time that the data is being accessed or read out for use elsewhere. The other operation is where new information is to be written into the storage cell. We have already illustrated the storage condition. Now let's take a look at typical read and write operations.

READ

Refer again to Figure 9-5. A read operation is initiated by applying a binary 1 to the word or row select line. This is done by the address decoder. This causes MOSFET switches Q5 and Q6 to conduct, which connects the flip-flop outputs from Q1 and Q3 to the DATA and $\overline{\text{DATA}}$ output lines.

Next, a binary 1 level is applied to the column or bit select control line. This too is done by the address decoder, which causes MOSFET switches Q7 and Q8 to be enabled. The flip-flop outputs are then passed through Q5 and Q6 as well as Q7 and Q8 to the inputs of a differential sense amplifier. The sense amplifier buffers the flip-flop output and generates a TTL compatible binary 0 or binary 1 output, depending upon the state of the storage latch.

The sense amplifier is usually buffered by a three-state control circuit. Recall that a three-state circuit can assume three different output conditions: binary 0, binary 1, and high impedance, or open. The chip select, or three-state control line in Figure 9-5 determines the state. If the chip select is binary 0, the data output will be either a binary 1 or a binary 0. If the chip select is a binary 1, the output is a high impedance or open circuit. In this condition, the sense amplifier output is essentially isolated or disconnected from the data output line. This permits the outputs of many static RAMs to be connected together or bussed. Most semiconductor memories are used in bus organized digital systems.

The three basic controls on the RAM are the row select, column select, and chip select. The input address is decoded and used to enable the appropriate row and column select lines to locate the desired bit or word in the memory. The chip select line is used to enable the memory IC where the desired data is stored. Usually, many chips are interconnected to form a complete semiconductor memory. The chip select lines are also controlled by decoders which are driven by the most significant bits of the address word.

WRITE

A write or store operation is initiated by applying the desired bit to the data input line. The data and its complement generated by the inverter are applied to write amplifiers.

Next, the row and column select lines are enabled by the address decoders. This causes Q5 through Q8 to conduct. The complementary outputs of the write amplifiers are then applied to the drains of Q1 and Q3, thereby forcing the latch to set or reset as desired. For example, assume that the output of write amplifier 1 is low and the output of write amplifier 2 is high. This causes the output of Q1 to be forced low and, as a result, the gate capacitance on Q3 is discharged. Q3 is, therefore, cut off. As a result, the gate capacitance of Q1 is charged through Q4 thus keeping Q1 on.

TTL RAMs

While most static RAMs are made with MOSFETs, bipolar static RAMs are also available. In fact, the earliest form of static RAM was a TTL device. MOSFETs are used in implementing RAMs because of their small size and low power consumption. This allows many data storage cells to be constructed on a single silicon chip. Bipolar devices take up much more space and consume a lot more power and, therefore, are less desirable for memories. However, the one thing going for bipolar memories is their very high speed. The read/write access time for bipolar memories are typically less than those of MOS memories. Therefore, in very high speed applications, bipolar memory devices are usually preferred. Both TTL and ECL RAMs are available.

Figure 9-7 shows the basic storage element of a TTL bipolar static RAM. It consists of two cross-coupled inverter transistors with multiple emitters. This basic flip-flop circuit latches into one of two possible states. The multiple emitters are connected to form row and column select lines as well as sense lines which are used for reading the output of the flip-flop or writing data into it.

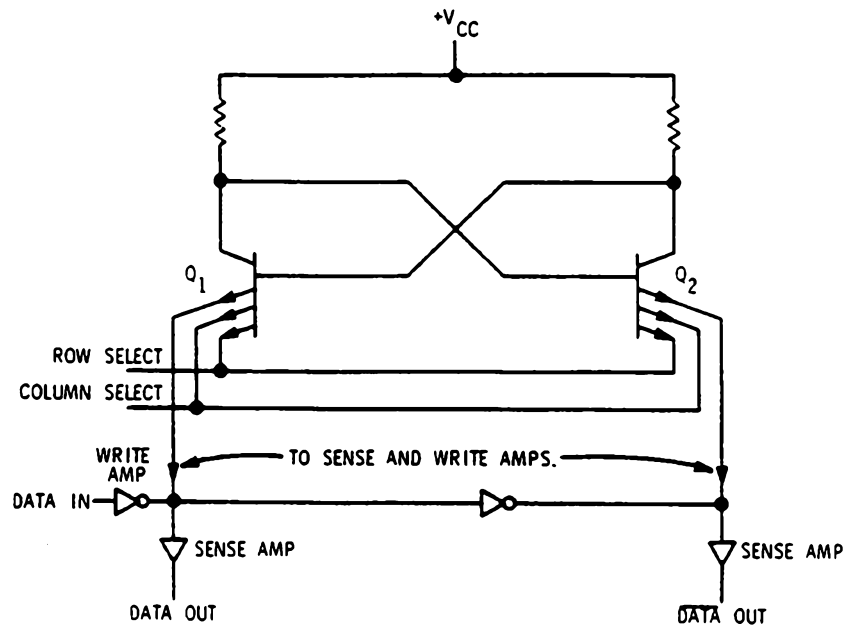


Figure 9-7
TTL storage cell.

The normal storage state of the flip-flop exists when both the row and column select lines are both at binary 0. The flip-flop is then in one of its two stable states. For example, assume that Q1 is conducting and Q2 is cut off.

To read the state of the flip-flop, both the row and column select lines are brought high or binary 1. This causes the conducting transistor Q1 to pass current through the low input impedance of the sense amplifier. The sense amplifier then generates the appropriate binary output state.

To write data into the cell, the row and column select lines are again brought to the binary 1 level. The flip-flop is then either set or reset by applying a binary 0 or binary 1 to the data input. This causes the emitter of Q1 or Q2 to be brought to ground. If the emitter of Q1 is grounded, a binary 1 is stored. The row and column select lines are then brought low. The latch then retains the state into which it is put. A RAM cell of this type has a typical access time of 20–50ns.

RAM Organization

Typical semiconductor memory ICs contain hundreds and even thousands of individual memory cells. Typical static RAM ICs are available for storing as many as 16K bits on a single chip. Other static RAMs are available in 256, 1K, 4K, and 8K bit configurations.

One of the most popular static RAMs is a 4K (4096) bit device. It is a good example of the typical organization of the storage cells and related circuitry in a static RAM IC.

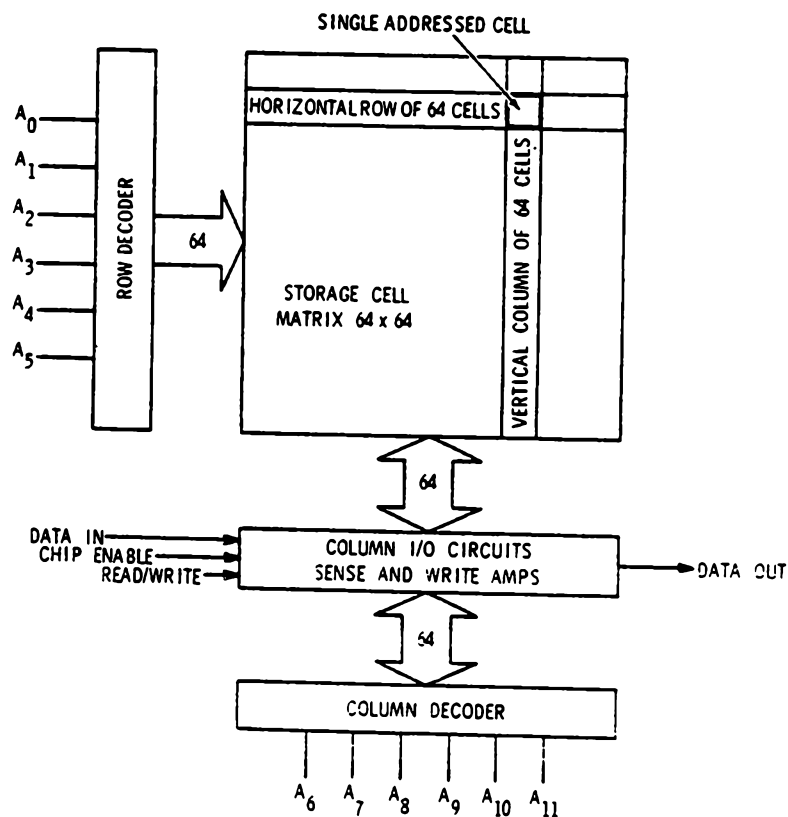


Figure 9-8

Typical organization of a 4K x 1 static RAM IC.

Refer to Figure 9-8. The 4096 memory cells are arranged in the form of a matrix with rows and columns. To store 4096 bits, the matrix has 64 rows and 64 columns. Note that there are both row and column decoders. These decoders accept the input address bits. To address 4096 bits, a 12-bit address word is needed. The address word is divided in half. The six lower order address bits A₀ – A₅ are applied to the row decoder while the six most significant address bits A₆ – A₁₁ are applied to the column decoder. With six input bits, each decoder produces 64 output lines. The 64 lines from the row and column decoders are used to enable one of the

storage cells in the matrix. Recall from the discussion of the static memory cell that row and column decode inputs are used to enable the various switches in the memory cell. The addressing method used in a static memory IC is one of X – Y coordinate selection. One of the 64 row decoder outputs will be high and one of the 64 column decoder outputs will be high for any given input address. These decoder outputs will then enable one of the 4096 bits in the memory matrix.

Once a particular memory cell has been addressed, it can then be read out or written into. The read and write operations are taken care of by the sense amplifiers and write amplifiers. Note in Figure 9-8 that the sense amplifiers and write amplifiers are associated with the columns of the matrix. There are two write amplifiers and one sense amplifier for each of the 64 columns in the memory. Refer back to Figure 9-5. All 64 static cells in one column are connected to a single sense amplifier and one set of write amplifiers. However, because the row and column decoders only enable one cell in the column, only that cell will use the sense and write amplifiers.

In addition to the individual data input and data output lines and the address lines on the RAM IC, there are two control inputs that specify the operation of the device. These are the read/write and the chip enable lines.

The chip enable (\overline{CE}) or chip select (\overline{CS}) is similar in some respects to an address line. This input line effectively enables the chip so that it is capable of performing either a read or a write operation. When the chip enable line is high, the IC is not selected. Even though there may be address inputs, they are effectively ignored and the chip performs no operation other than retaining data previously stored there. With the chip enable high, the data input line is electrically disconnected from the input circuitry to the storage cells. The data output which is derived from a three-state device is placed into its high impedance state. The output pin simply appears to be an open circuit. The chip is effectively disabled at this time.

When a chip enable input is brought to a binary 0, the chip is selected. Data input signals are recognized and the data output buffer is enabled so that the selected memory cell state is presented at the output.

The read/write input, also called write enable, specifies the operation the chip is to perform. With the R/W line high, the IC performs a read operation. The bit stored in the cell selected by the input address is presented at the data output pin. A write operation is performed when the R/W line is low. A bit to be stored is applied to the data input pin. That pin is stored in the cell selected by the address lines. Incidentally, this basic matrix organization is valid for dynamic as well as static RAMs.

Figure 9-9 shows the timing waveforms for both the read and write operations of a typical 4K static RAM.

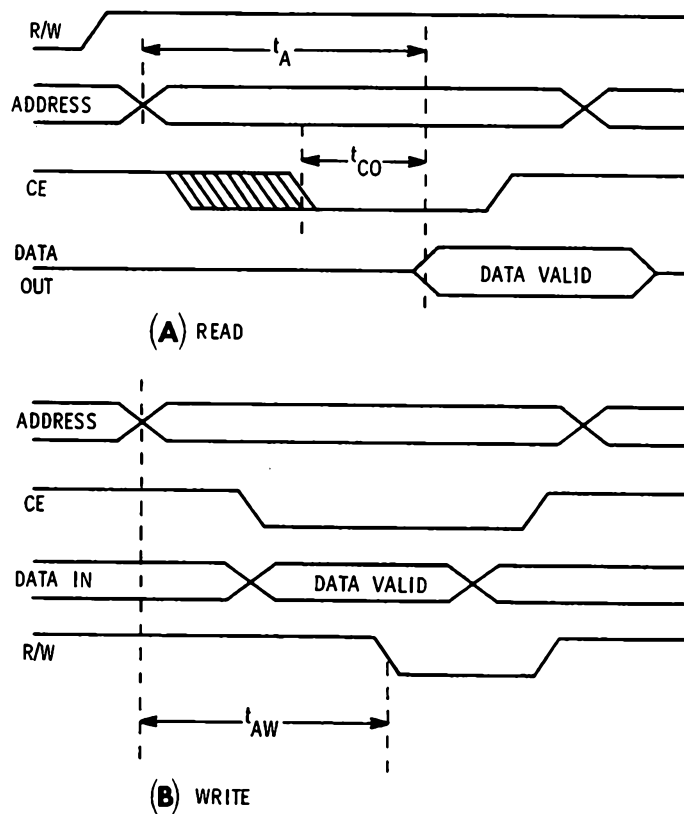


Figure 9-9
Static RAM timing (A) read (B) write.

Refer to Figure 9-9A. To perform a read operation, the R/W line is made high. The desired address is then applied to the chip. In addition, the chip enable (CE) line must be brought low before data appears at the output pin. The timing diagram shows the chip enable as a pulse that goes low a short duration after the address lines change. When the chip enable goes low, data will appear at the output for a short time thereafter. This time is designated as t_{CO} in the diagram. The timing relationship between the input address and the chip enable is not critical. Therefore, it is possible to simply tie the chip enable input low to permanently enable the chip. The device can then be operated with the address inputs and the CE line. By keeping the R/W line high, the input addresses may be changed in any order to access data at random. With the chip enable low and the R/W line high, the data appears at the output a time duration t_A after the input address lines change. This is the access time. Typical access times for a MOS static RAM device of this kind is 50 to 300 nanoseconds.

To perform a write operation, the desired address is applied to the chip. The chip enable is also brought low and input data is then applied. The R/W line is then brought low and the data is stored. This timing sequence is illustrated in Figure 9-9B. If it is necessary to provide a continuous series of write operations, the chip enable line can be held low while the addresses are changed. However, the R/W input must be timed in such a way that it occurs after the input address change has occurred. A minimum write to address time set-up, t_{AW} , must be observed for proper operation. This means that the R/W line must not go low for a period of approximately 20 nanoseconds or more after the input address lines settle into the desired state.

Alternate Architectures

The static RAM we have been discussing is organized as a $4K \times 1$ storage device. This means that there are 4096 individual locations for storing a single bit word. However, there are other organizations that are also useful. Another common way of organizing a 4K memory is into 1024 four bit words. This $1K \times 4$ organization still uses the $64 \text{ row} \times 64 \text{ column}$ matrix. To address 1K words, a 10-bit address word is used. The six most significant address bits are applied to the row decoder which in turn selects one of the 64 internal matrix rows 0 – 63. The four most significant address bits are applied to the column decoder. While there are 64 columns in the matrix 0 – 63, these will not be enabled individually. Instead, four columns will be enabled simultaneously.

R/W Memories

The four inputs to the column decoder are translated by the decoder into sixteen output lines which enable sixteen groups of four columns each in the 64×64 matrix. See Figure 9-10.

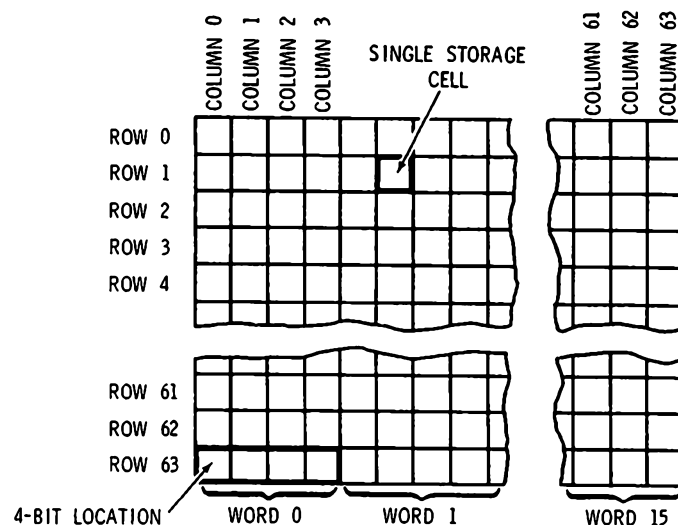


Figure 9-10
Storage cell matrix organization in a $1K \times 4$ static RAM.

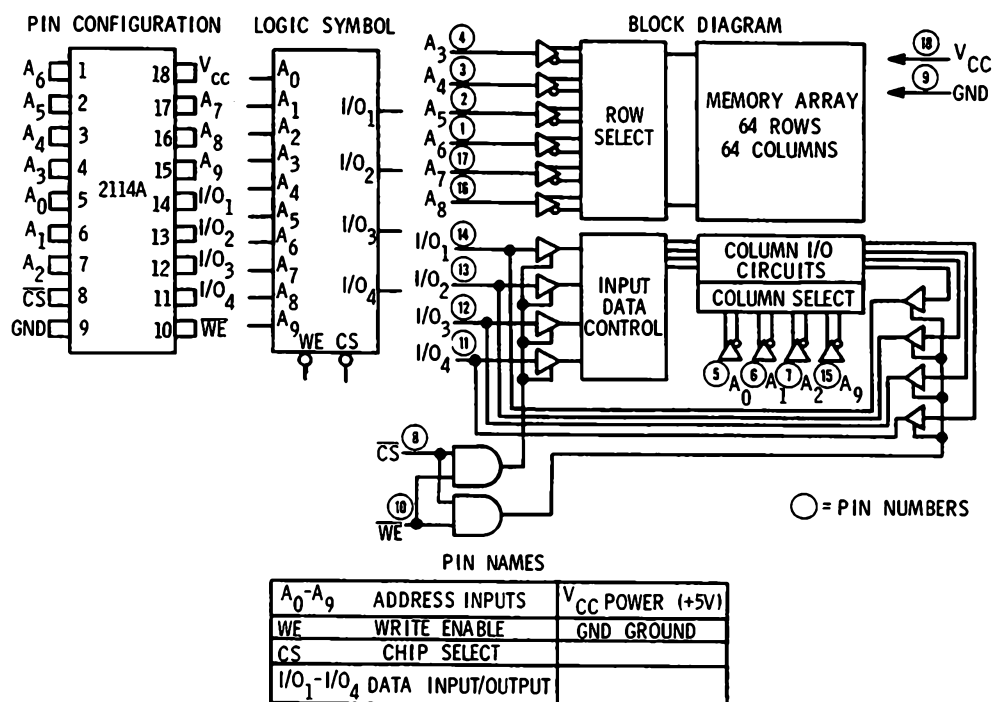


Figure 9-11
The popular 1K × 4 static RAM.

Figure 9-11 shows a general block diagram of the 2114, a typical 1K × 4 static RAM. In addition to the 10 input address lines A₀ – A₉, there are chip select (\overline{CS}) and write enable (\overline{WE}) input control lines. A common set of data input/output (I/O) lines are also provided. When a read operation is being performed, the data stored in the memory is presented on the I/O lines. To store a 4-bit word in memory, that word is applied to the four I/O lines. The \overline{WE} control determines whether these lines are to be used for input or output operations. Aside from this difference and the basic internal organization, the 1K × 4 device performs in exactly the same way as the 4K × 1 device discussed earlier.

The 2114 1K × 4 static RAM is housed in an 18-pin DIP. Note the logic symbol used to represent it. You will use a 2114 RAM in Experiment 23.

Typical Memory Configurations

A single RAM chip is rarely, if ever, sufficient in itself to provide all the memory required by a given application. For that reason, many memory chips are interconnected to form complete memory circuits. A couple of examples using the $4K \times 1$ and $1K \times 4$ memory chips discussed earlier will illustrate this concept.

Refer to Figure 9-12. Here sixteen $4K \times 1$ ICs are interconnected to form a $16K \times 4$ memory system. The $4K \times 1$ chips are interconnected into four groups of four similar circuits. Each four bit group is capable of storing 4096 four bit words.

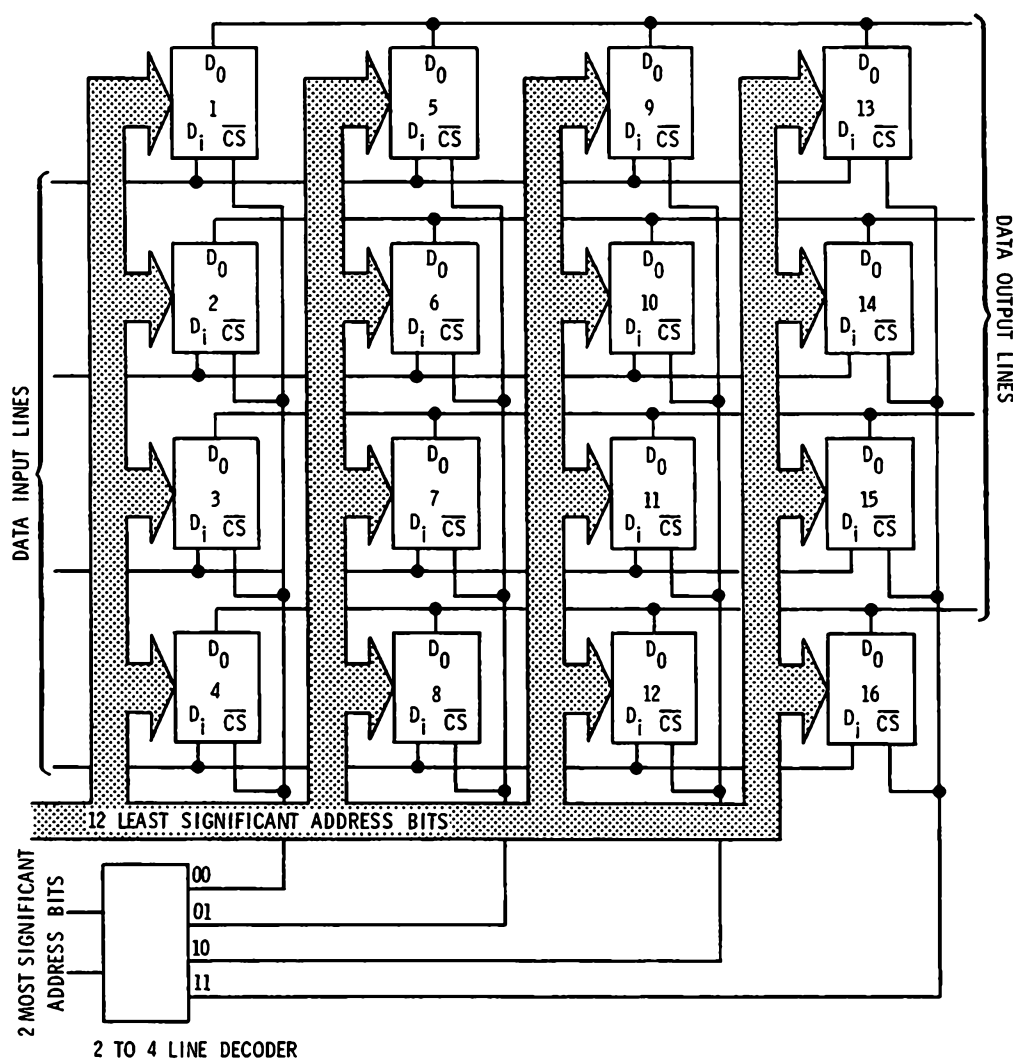


Figure 9-12

A $16K \times 4$ memory made up of $4K \times 1$ chips.

To address 16K words, a 14-bit address is needed. The twelve least significant bits of the address are applied to all of the 4K memory chips simultaneously. The remaining two most significant bits are applied to a 2 to 4 line decoder. The four decoder outputs are applied to the chip select (\overline{CS}) inputs of each of the four groups. Note that the data input (D_i) and data output (D_o) lines are connected together. The data output lines are three-state in nature and only one of the four chips interconnected in this way is enabled at a time. All of the read/write lines on each chip are interconnected. To simplify the circuit, however, the R/W lines are not shown.

To illustrate how the circuit works, assume that we apply the address word 01000000000000. The 12 least significant bits are all zeros and are applied to all sixteen memory chips. This means that storage cell 0 in each chip is enabled. The two most significant bits are 01. This causes the 01 output line of the decoder to be enabled. This line goes low, enabling chips 5 through 8 in the circuit. Chips 1 through 4 and 9 through 16 are disabled at this time. If a read operation is to be performed, the data output lines of chips 5 through 8 will assume the states of the zero cells in each of those chips.

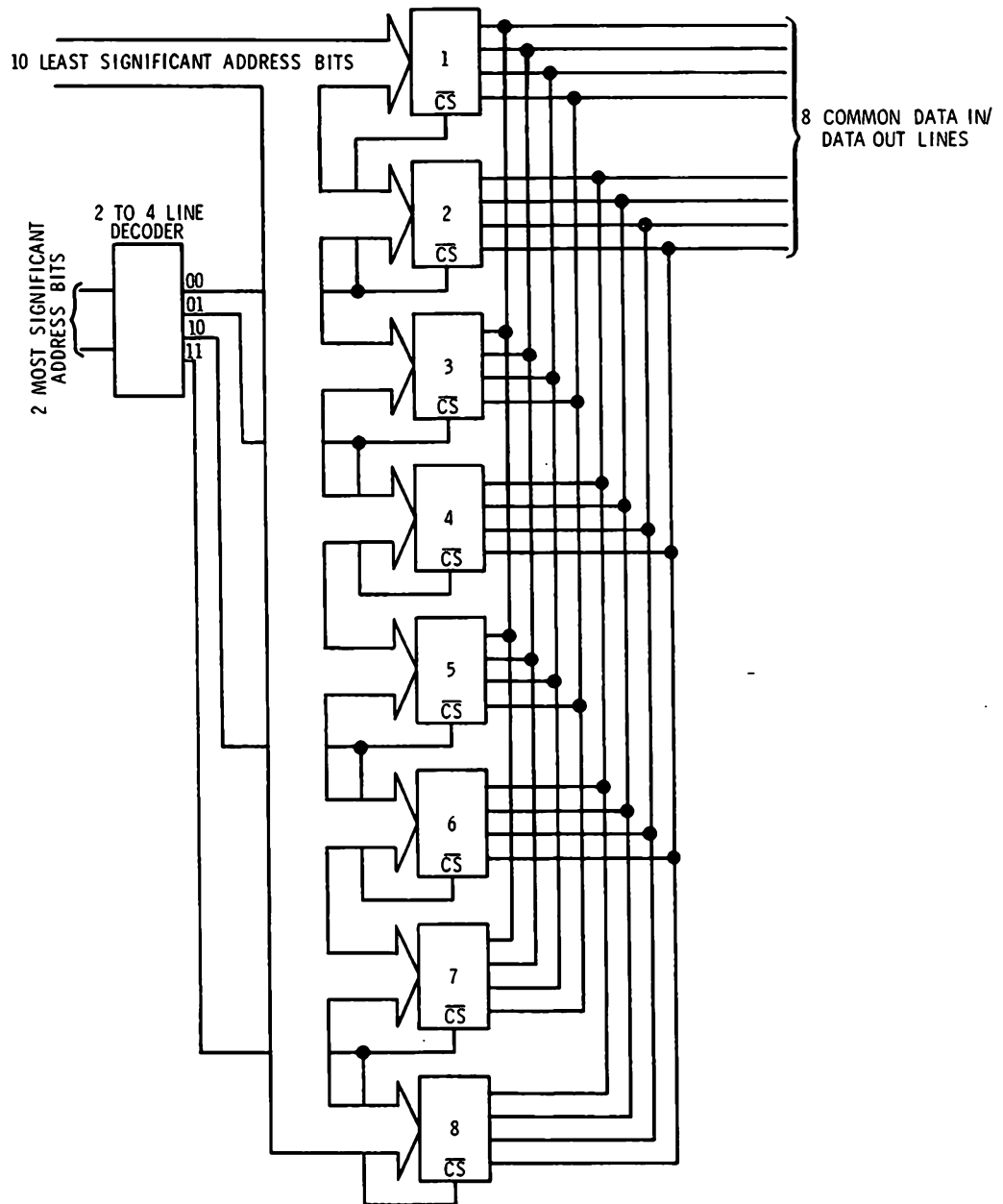


Figure 9-13
A $4K \times 8$ memory made with $1K \times 4$ chips.

Figure 9-13 shows how $(1K \times 4)$ 2114 RAM ICs are interconnected to form a $4K \times 8$ memory. To address 4K of memory requires a 12-bit address. The 10 least significant bits of address are applied to all eight memory chips simultaneously. The two most significant bits are applied to a 2 to 4 line decoder. The decoder outputs are used to select one of four groups of 1K words of memory. This is done by tying the decoder inputs to the appropriate chip select (\overline{CS}) inputs. ICs 1 and 2 contain the first 1K bytes of memory, four bits in each chip. Chips 3 and 4 contain the second 1K bytes while chips 5 and 6, and 7 and 8 contain the third and fourth 1K bytes of data. The three-state outputs are connected together to form two four bit groups of outputs. Remember that the data in and data out pins on 2114 ICs are common. The read/write lines are not shown, but are all interconnected.

To show how the circuit works, assume that the input address is 110000000000. The ten least significant bits are all zeros, so the first four cells in each memory chip are selected. The two most significant bits 11, causes the 11 line of the decoder to go low. This enables chips 7 and 8. All other decoder outputs are high at this time, therefore, chips 1 through 6 are disabled.

Self Test Review

22. The two types of read/write memories are _____ and _____.
23. A read/write memory is called a _____.
24. The main storage element in a static RAM is a _____.
25. In Figure 9-5, the storage latch loads are _____.
26. Faster storage cell operation is obtained by using _____ loads.
27. To address a storage cell the _____ and _____ select inputs must be enabled.
28. TTL static RAMs are slower than MOS RAMs.
 - A. True
 - B. False
29. A typical static RAM chip has a maximum storage capacity of _____ bits.
30. RAM storage cells are arranged in the form of a _____ with multiple _____ and _____.

31. In a $4K \times 1$ RAM, only one of the 4096 bits is addressed at a time.
- A. True
 - B. False
32. A RAM chip is selected if the chip enable line is:
- A. Binary 0
 - B. Binary 1
33. The 2114 RAM has the following organization:
- A. $1K \times 1$
 - B. $4K \times 1$
 - C. $1K \times 4$
 - D. $4K \times 4$
34. All of the bits in a column share the same sense and write amplifiers.
- A. True
 - B. False

Answers

- 22. static, dynamic
- 23. RAM
- 24. flip-flop
- 25. MOSFETs
- 26. depletion mode
- 27. column, row
- 28. B — False. Bipolars are faster than MOSFETs.
- 29. 16K
- 30. matrix, rows, columns
- 31. A — True
- 32. A — Binary 0
- 33. C — $1K \times 4$
- 34. A — True

DYNAMIC MEMORIES

The most widely used form of semiconductor memory is the dynamic type. While static memories are easy to understand and apply, they do have disadvantages for some applications. Compared to dynamic memories, static memories cost more per bit of storage. In addition, static memories consume more power.

The primary reason for the high cost per bit of a static memory is the relatively large amount of area required to make a static cell on a silicon chip. Most static storage cells consist of from 4 to 8 MOS transistors per bit. As you might suspect, this takes up a considerable amount of area despite the small size of the typical MOSFET device. Because of the large number of devices required to make a cell for a given size silicon chip, the number of total bits of storage is limited. IC cost is directly related to chip area. Static memory circuits also have relatively high power consumption. When a large volume of storage is required, static memory devices are less desirable because of the need for high current power supplies and expensive cooling systems.

Dynamic semiconductor memories overcome these disadvantages. The typical dynamic memory storage cell is simpler and occupies considerably less chip area. Approximately four times as many dynamic cells than static cells can be made on a silicon chip of a given size. Dynamic memory cells also have lower power consumption than static cells. The resulting lower cost per bit and lower power consumption make dynamic memories extremely popular in digital designs. In fact, they are almost universally used in all modern digital computers from the smallest microcomputer to the largest, most expensive mainframe computer.

But while the dynamic memory is more widely used than static devices, they are not without their problems. Dynamic semiconductor memories are far more complex in their operation and application. They require special support circuitry and often have critical timing requirements. Dynamic memories are also slower than static memories. The higher access times make dynamic cells less desirable than static cells for some applications. Nevertheless, the semiconductor manufacturers have made these disadvantages easy to deal with.

DYNAMIC CELLS

The basic storage element in a dynamic memory cell is a capacitor. The capacitor can be either a tiny discrete integrated capacitance or the gate to channel capacitance of a MOSFET. When the capacitor is charged, it is storing a binary 1. When the capacitor is discharged, it is storing a binary 0.

The capacitance of a dynamic cell is typically in the very low picofarad range. The actual capacitance value varies considerably with the dielectric used in making the capacitor and the physical dimensions of its plates. The characteristics of the MOSFET also affect the capacitance. But, despite their small size, these tiny integrated capacitors are capable of storing a charge long enough to make them practical as a storage element. However, shunt leakage resistance does cause the charge to leak off after a short period of time. Because of the small capacitance and the leakage resistance, it is necessary to periodically restore the charge on the capacitor to prevent the loss of data. This process of recharging the storage cell is called refresh. A refresh operation is performed each time a dynamic memory cell is read. In addition, the memory cell is refreshed at a regular periodic rate, determined by a clock circuit, as long as power is applied to the memory. Most dynamic memories are refreshed approximately every 2 milliseconds to avoid the loss of data. The necessity for the refresh operation is one of the disadvantages of a dynamic cell over a static cell. The refresh circuitry and timing operations are critical to the proper operation of a dynamic memory. Should the refresh operation not be performed, data will be lost. Of course, dynamic semiconductor memory cells are also volatile as data will be lost if power is removed.

THREE TRANSISTOR CELL

Now let's take a look at the details of some typical dynamic memory cells. One of the earliest dynamic storage cells is shown in Figure 9-15. This basic cell uses three MOSFETs. The basic storage element is the gate capacitance of Q2. Q1 and Q3 are MOSFET switches that are used to control read and write operations.

To perform a write operation, the bit to be stored is applied to the data input line. The write enable line is then triggered. This causes Q1 to conduct, and the bit to be stored is applied to the gate of Q2. Depending upon whether the input is a binary 1 or a binary 0, the gate capacitance of Q2 is either charged or discharged.

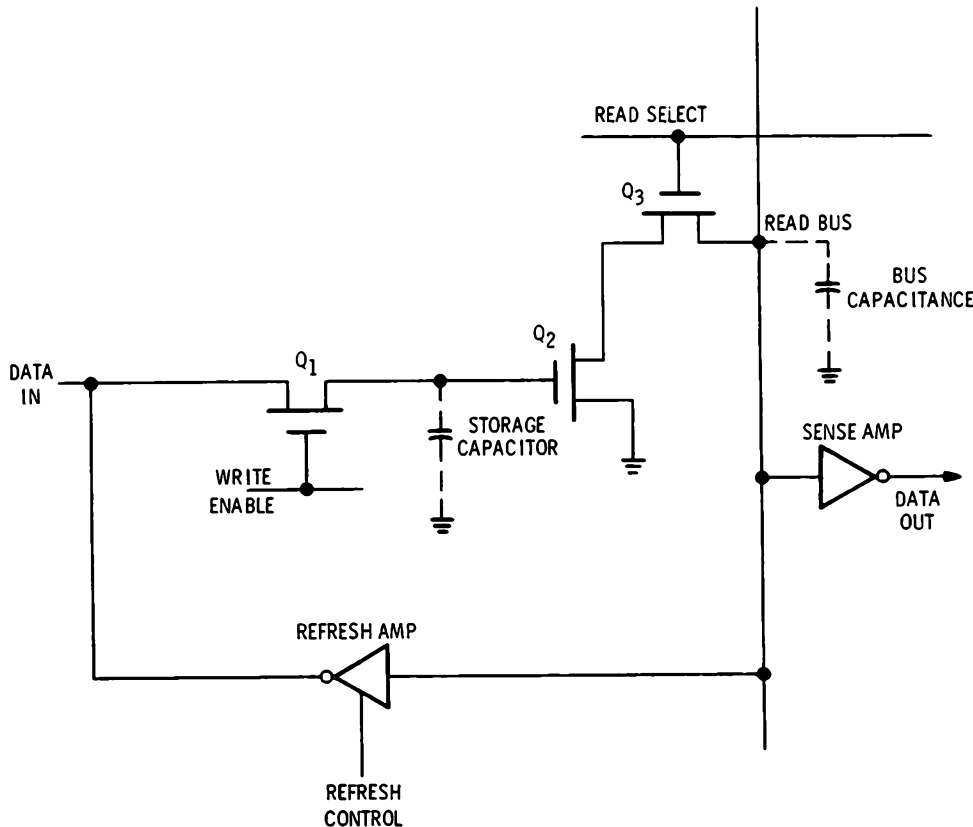


Figure 9-15
Three transistor dynamic storage cell.

To perform a read operation, the read bus capacitance is precharged. The read bus is a common line to which many memory cells are connected. This bus has an inherent distributed or parasitic capacitance. Circuitry within the chip is used to charge the bus capacitance.

An address decoder circuit then enables the read select line and causes Q3 to conduct. With Q3 conducting, Q2 is connected to the read bus. If a binary 1 happens to be stored in the storage cell, the gate capacitance of Q2 will be charged. As a result, Q2 will conduct and act as a low, which causes the read bus capacitance to discharge. This reduces the voltage on the read bus by a predetermined amount. The voltage level on the read bus is sensed by a sense amplifier which switches and generates the appropriate binary level at the data output. If a binary 0 is stored in the cell, the gate capacitance of Q2 will be discharged and Q2 will not conduct. Its high impedance will not affect the read bus capacitance, so the charge on the read bus capacitance will remain high. The output sense amplifier detects the higher level and generates the output voltage level for a binary 0.

Now let's consider the refresh process. Typically the MOSFET gate capacitance and its associated leakage resistance is such that a charge can be retained for several milliseconds. However, if that charge is not refreshed, the data will be lost. To restore the data, a refresh operation is performed by reading the contents of the cell and then writing it back in.

The refresh operation of the cell in Figure 9-15 is initiated by first charging the read bus capacitance. The read select line is then enabled by the address decoder, which causes Q3 to conduct and connects Q2 to the read bus. As indicated earlier in the description of the read operation, Q2 will either conduct or not conduct depending upon whether its gate capacitance is charged or discharged. The voltage on the read bus is detected by the output sense amplifier and the appropriate output bit is generated. Note in Figure 9-15 that the read bus voltage is also sensed by the refresh amplifier. Like the output sense amplifier, the refresh amplifier detects the voltage on the read bus and generates an appropriate output bit. Note that the output of the refresh amplifier is applied back to the data input line.

The refresh cycle is complete when the write enable line is enabled. This causes Q1 to conduct and apply the output of the refresh amplifier to the gate of Q2, either charging or discharging its gate capacitance. This refresh process takes place approximately every 2 milliseconds in most dynamic memories. Keep in mind that a refresh operation also generally takes place during a read operation.

ONE TRANSISTOR CELL

The basic three transistor dynamic cell shown in Figure 9-15 was most widely found in the popular 1103 dynamic $1K \times 1$ memory chip. This device, introduced in the early 1970's, was one of the earliest and most versatile dynamic chips. Since that time, developments in dynamic cell circuitry and semiconductor technology now permit many more bits of data to be stored on a single chip. Today, 4K, 16K, and even 64K bit dynamic chips are available. Such very high density memories are the result of an ultra simple dynamic memory cell consisting of a single capacitor and a single MOSFET switch.

A typical single transistor dynamic storage cell and its associated support circuitry is shown in Figure 9-16. The main storage element is a tiny integrated capacitor CS1 or CS2. A single MOSFET switch (Q1 or Q2) connects the capacitor to the bit sense line for both read and write operations. Note that there are two bit sense lines, one on each side of the sense amplifier/write latch (Q6 and Q7) lines, each with many storage capacitors and switches connected to them. For example,

in a $4K \times 1$ dynamic RAM, there would be 64×64 storage cell matrix. In Figure 9-16, 32 storage cells would appear on each side of the sense amplifier/write latch. This would represent one 64-bit column. The individual cells are enabled by row-select lines from the row decoders.

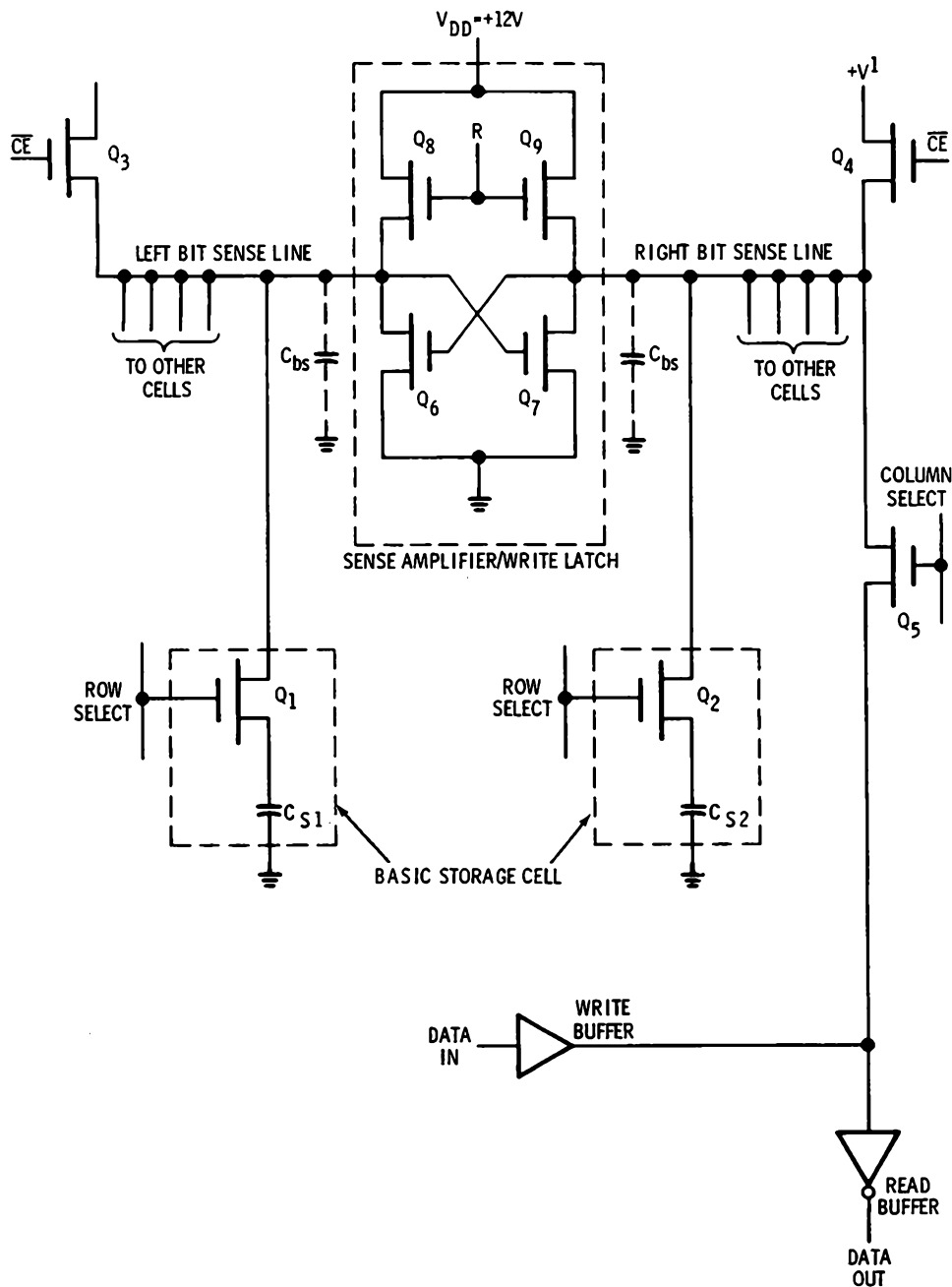



Figure 9-16
Typical single transistor dynamic storage cell and support circuitry.

To write data into a cell, the address is applied to the chip. The row and column decoders enable the desired cell. Assume Q1 and Q5 are turned on in Figure 9-16. Assume that a binary 0 is to be stored and is applied to the data input. The output of the non-inverting write buffer is a binary 0. This is passed through Q5 to the right bit sense line. This grounds the drain of Q7 and the gate of Q6. Q6 is cut off and Q7 is biased on. The sense amplifier/write latch locks-up in that state. The left bit sense line is high or binary 1, so storage capacitor CS1 is charged through Q1 and the data is stored. Note that the input binary 0 is stored as a binary 1.

Now assume that we had stored the binary 0 in storage cell CS2 rather than CS1. The input would still cause the sense amplifier/write latch to lock up in the same state as before. The right bit sense line would be low and cell CS2 would be discharged when Q2 conducts. As you can see, data is stored in its normal state in those cells on the right bit sense line and in the complement state on the left bit sense line. The reason for this, of course, is the complementary outputs of the sense amplifier/write latch. It doesn't really matter how the data is stored as it will come out correctly when it is read.


A read operation is performed as follows. Refer again to Figure 9-16. When the RAM chip is not selected, \overline{CE} is high, and causes Q3 and Q4 to conduct. The bit sense line capacitances C_{bs} are charged to some value of voltage between +VDD (usually 12 volts) and ground. This would seem to force the sense amplifier/write latch into an ambiguous state, but it doesn't because load transistors Q8 and Q9 are cut off by the low R (Read) signal applied to their gates.

When the chip is selected, due to \overline{CE} going low, Q3 and Q4 are cut off and open. Next, the address is applied, turning on the MOSFET associated with the desired cell (represented by either Q1 or Q2 in Figure 9-16). Q5 also conducts. Next, the R (Read) signal is applied to Q8 and Q9. This enables the latch, which essentially compares the right and left bit sense lines. If storage cell CS2 was chosen, Q2 would be conducting. The binary 0 previously stored there means that CS2 is discharged. With Q2 on, CS2 and the bit sense capacitance C_{bs} are connected in parallel. C_{bs} discharges while CS2 charges. This causes the right bit sense line to go low and makes the left bit sense line high. Therefore, the sense amplifier/write latch locks up with Q6 off and Q7 on. With Q7 on, the right bit sense line is low. This is passed through Q5 to the inverting output buffer. The data output is a binary 1. A binary 0 was stored in CS2 as a binary 0 but is read out as a binary 1. The output of a dynamic RAM is usually the complement of the bit stored.



If we had tried to read cell CS1, Q1 would conduct. We stored a binary 0 there previously. The binary 0 input was retained in CS1 as a binary 1. CS1 was charged. During the read operation, Q1 conducts, connecting CS1 in parallel with C_{bs} . The voltage on CS1 is greater than that on C_{bs} of the left bit sense line. Therefore, C_{bs} charges while CS1 discharges. The voltage on the left bit sense line is higher than that on the right bit sense line. This causes the latch to switch so that Q6 is cut off and Q7 conducts. This is exactly the same condition as when the original binary 0 was stored in CS2. The right bit sense line is low, making the data output a binary 1. As you can see, a bit can be stored as a binary 0 or binary 1 depending on which side of the sense amplifier/write latch the desired cell is located. But, when a read operation is performed, the correct state is returned. It just so happens that the correct state is the complement of the originally stored bit.

A refresh operation is performed each time a cell is read. Reading a cell is destructive as the state of the storage cell capacitor is transferred to the bit sense line capacitance. But as soon as the sense amplifier/latch changes state and stores the bit, its output has the correct state. The storage cell capacitor then charges or discharges to this state, refreshing the data.



A refresh operation is performed every 2 milliseconds to ensure that no data is lost. This is done by simply initiating a read cycle.

Dynamic RAM Organization

Almost all dynamic RAMs are organized as multiple one bit storage locations. Typical dynamic RAM organizations are $16K \times 1$, $64K \times 1$, $256K \times 1$, $512K \times 1$, and $1M \times 1$. These RAMs are usually housed in a standard dual inline or surface mount IC package. Multiple ICs are then interconnected to form multi-bit memories capable of storing thousands of 8-, 16-, or 32-bit words.

Internally, the dynamic memory cells are organized into a row and column matrix like that described earlier in the section on static memories. However, because of the higher density of dynamic memories, there are variations in the row and column matrix arrangement. A number of different methods of partitioning the memory cells are used to simplify the design and minimize circuitry. This is particularly true in the higher density dynamic RAM chips.

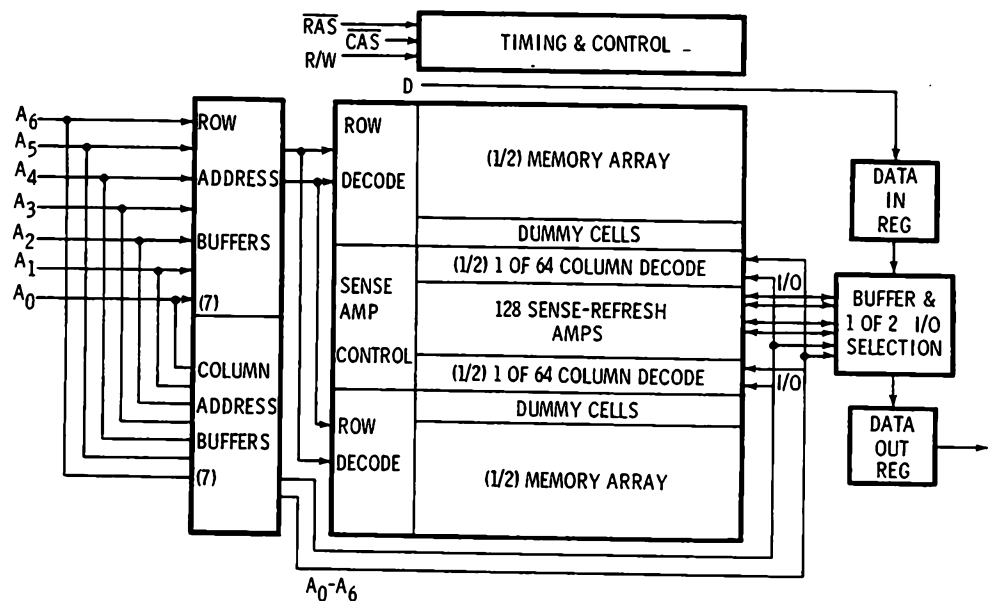


Figure 9-17
Block diagram of the 4116 $16K \times 1$ dynamic RAM.

Now let's take a look at a typical dynamic RAM IC. Figure 9-17 shows a simplified block diagram of the 4116 $16K$ dynamic RAM. The 4116 is the most popular and most widely used $16K$ RAM. It is about as close to an industry standard as there is. It uses the common $16K \times 1$ organization. The standard package is a 16-pin DIP.

The dynamic storage cells are arranged in a modified 128×128 matrix ($128 \times 128 = 16,384 = 16K$). This matrix is split into two groups of 64 rows by 128 columns. This particular arrangement is used because it gives a simplified physical chip layout. Tiny high speed digital circuits are subject to noise and cross talk. Therefore, physical layout is extremely important in the design to minimize such difficulties.

Both row and column decoders are used to address one of the 16,384 bits. The row decoders are divided into two groups and the column decoders are divided into two groups. Note that there are 128 sense and refresh (write) amplifiers, one for each column.

It takes 14 bits to address 16,384 bits ($2^{14} = 16,384$). However, note that the 4116 RAM has only 7 input address bits (A0-A6). In order to reduce the number of pins on the IC package, only half the required address bits are used. The 14-bit address is applied to the internal memory matrix in 7-bit groups. The 7 lower order bits are first applied to the address lines and are stored in the row address buffers. The control signal \overline{RAS} , which means row address strobe, stores the 7 least significant bits in the row address buffers when it switches from high to low. The 7 higher order bits are then applied to the IC pins. These are then stored in the 7-column address buffers when the column address strobe (\overline{CAS}) line switches from high to low. The outputs of the row and column buffers, or latches, drive the row and column decoders which then select one of the 16,384 bits.

The only other control line on the IC is a read/write (R/\overline{W}) line. When the R/\overline{W} line is high, a read operation is performed. When the R/\overline{W} line is low, a write operation is performed. Separate single bit data input and data output lines are provided. The output is derived from a three-state buffer so these ICs may be used in bus organized systems.

READ OPERATION

Figure 9-18 shows the timing diagram for a read operation on the 4116 RAM chip. The 7 lower order bits of the address are applied to pins A0 through A6. These are strobed into the row address latches by the external signal $\overline{\text{RAS}}$. Next, the 7 higher order bits of the address are applied to pins A0 through A6. These are strobed into the column address latches by signal $\overline{\text{CAS}}$. The $\text{R}/\overline{\text{W}}$ line can go high after $\overline{\text{CAS}}$ to enable the read function. This takes the output buffer out of its high impedance state so that a binary 1 or binary 0 is read out; depending upon the bit stored in the address locations. The selected memory location will present data to the output pin approximately 100 nanoseconds after $\overline{\text{CAS}}$ goes low. Both $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ can go high as soon as the data is valid. In this case, the data will remain at the output for a period of approximately 40 nanoseconds after $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ go high. The $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals can be kept low for an extended period of time so that data will be available at the output longer. However, this extended period of time may not be more than approximately 10 microseconds. Otherwise, the memory will not be refreshed and data will be lost.

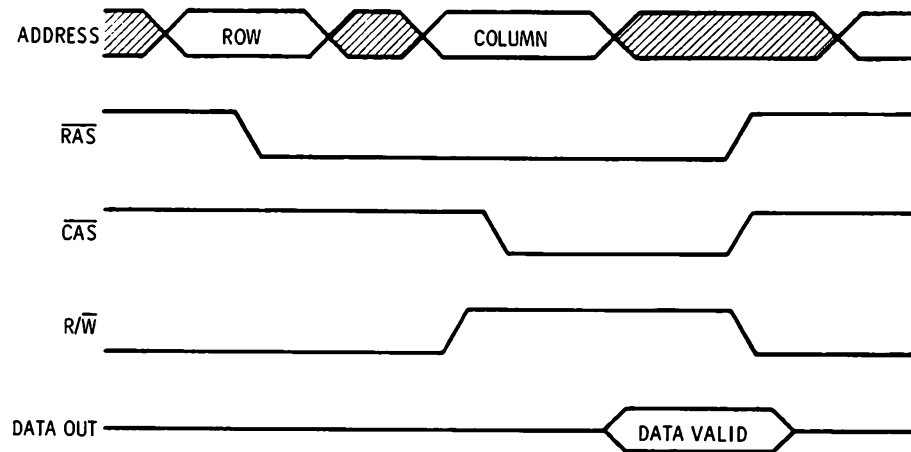


Figure 9-18
Read cycle of 4116 dynamic RAM.

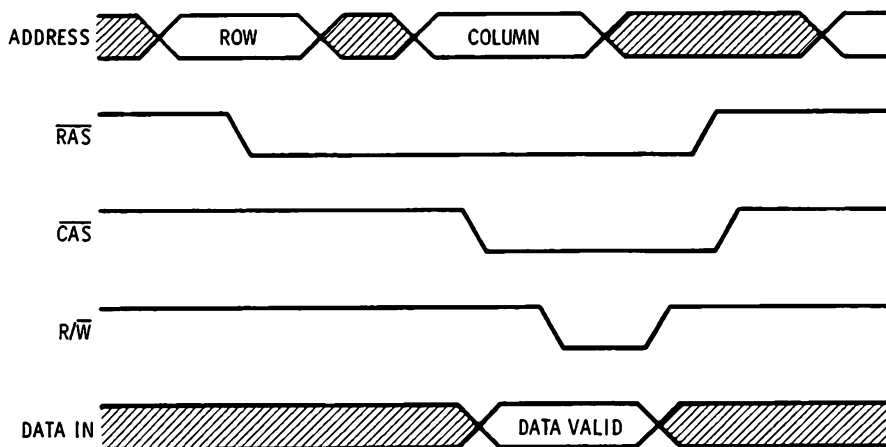


Figure 9-19
Write cycle of 4116 dynamic RAM.

WRITE OPERATION

Refer to Figure 9-19. As in the read operation, the desired memory cell is first addressed by applying the 7 lower order bits and strobing them into latches with $\overline{\text{RAS}}$. The higher order bits are strobed into the column latches by $\overline{\text{CAS}}$. The bit to be stored is applied to the data input. As soon as the address has been strobed, the $\overline{\text{R/W}}$ input line can be brought low, thereby storing the input bit in the addressed location. During the write operation, the data output line is in its high impedance state.

REFRESH

In order for data to be retained in the memory, a refresh operation must be performed every 2 milliseconds or less. In most dynamic RAMs this is done on a row by row basis. In the 4116, the row address latches are sequentially loaded with the 7-bit address for each row in the memory. This is usually done by connecting a 7-bit binary counter to the RAM address input pins. The counter is incremented and each address is strobed into the row address latches by the $\overline{\text{RAS}}$ input signal. Each time $\overline{\text{RAS}}$ goes from high to low, all 128 bits in the address row are read out and rewritten. Since each row must be refreshed every 2 milliseconds, and since there are 128 rows, then the time interval between sequential row refreshes is $2\text{ms}/128 = 15.625$ microseconds. This refresh operation goes on continuously as long as power is applied to the dynamic RAM chip.

The refresh operation and related control functions are generally performed by a special integrated circuit known as a dynamic RAM controller. A general block diagram of such a device is shown in Figure 9-20. The main function of this chip is to perform various multiplexing operations. One multiplexing operation performed by this chip is that of supplying first the lower order address bits and next the upper order of address bits to the RAM chip. Usually, all 14 address bits are available in parallel simultaneously. A multiplexer circuit is required to feed the two halves to the RAM chip one after the other.

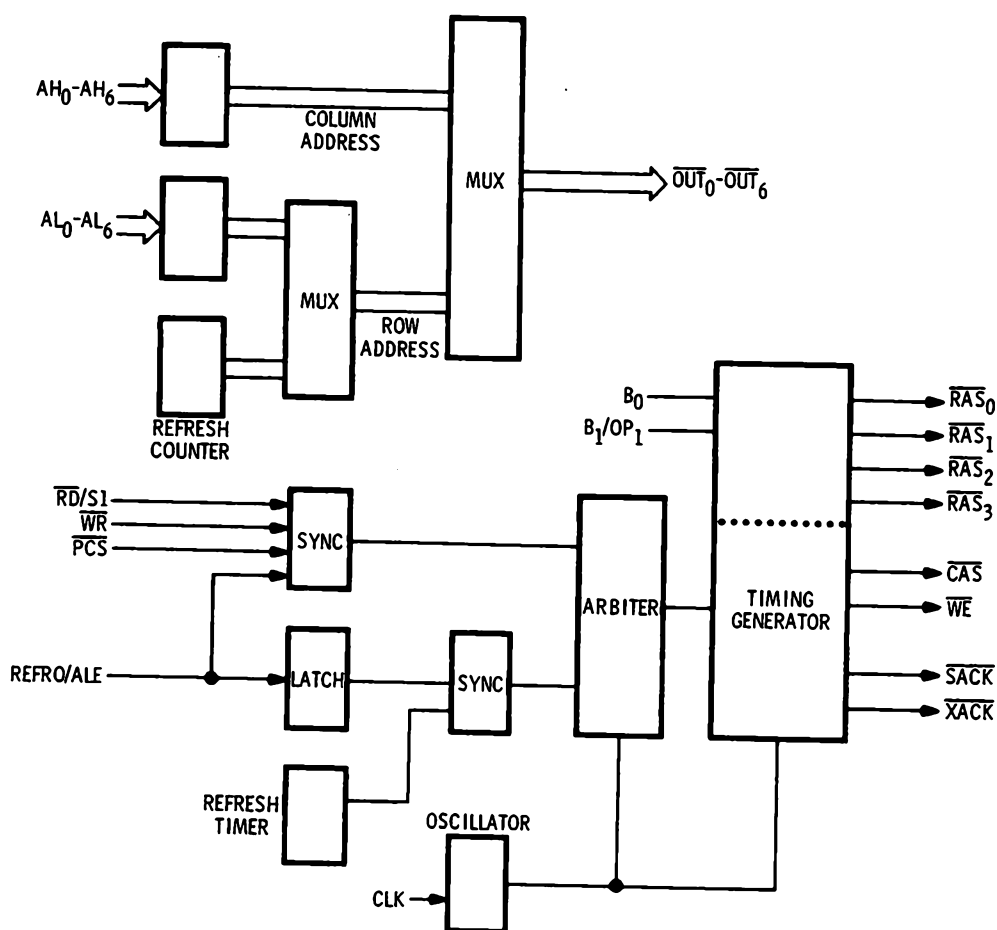


Figure 9-20
Dynamic RAM refresh controller.

Another multiplexing operation performed by this chip is the multiplexing of the 7 lower order address bits and the 7 input bits from the refresh counter. The 7-bit refresh counter is contained within this chip. The output of the multiplexer circuit is applied to the 7 address pins on the RAM chip.

The dynamic RAM controller chip also provides \overline{RAS} , \overline{CAS} , \overline{WE} control signals for the RAM. These are produced by the timing generator in Figure 9-20. The refresh timer is used to monitor the elapsed time since the last refresh cycle occurred. When a specific amount of time elapses, the refresh timer will request a refresh cycle. An external refresh request input is provided so that refresh cycles may be initiated from another source. If this input is not used, the refresh timer provides the necessary time information to ensure that the refresh operation takes place on time to avoid the loss of data.

An internal oscillator provides the basic timing signal for stepping the refresh counter and operating the other timing circuits in the controller chip. An external clock input signal is provided so that a separate system clock may be used to initiate refresh and other control operations.

The arbiter circuit is designed to resolve conflicts between refresh and memory requests. When read and write operations are not requested, the arbiter simply senses the inputs from the refresh timer and initiates the appropriate refresh cycles. If a read or write operation is requested, the arbiter gives the read or write request priority over the internal refresh request. Keep in mind that when a read operation is initiated, a refresh is generated automatically.

64K and Larger RAMs

The organization and operation of a typical 64K bit RAM is virtually identical to that of the 16K RAM. Most 64K RAMs are organized as 64K or 65,536 locations for one-bit words. 16 bits are required to address 64K locations. Eight address lines are provided on the 64K RAM chip. The two 8-bit portions of the address are multiplexed on these lines as they are in the 16K device. All other operations, read, write, and refresh are similar to the 16K device. Typical access times are 60 to 200 nanoseconds.

While the 64K dynamic RAM is very popular and widely used; 256K, 512K, and 1M RAMs are also available. These RAM chips with their enormous storage capacity may not be too practical for some applications. However, they do find use in 16- and 32-bit microcomputers, minicomputers, and mainframe computers. Of course, the semiconductor manufacturers will continue to improve the technology and make even higher density RAMs available in the future.

Self Test Review

35. For a given chip size, more bits can be stored in a:
 - A. Static RAM.
 - B. Dynamic RAM.
36. Which RAM is faster?
 - A. Static.
 - B. Dynamic.
37. Which RAM has lowest power consumption?
 - A. Static,
 - B. Dynamic.

38. The popular 4116 RAM has which organization?
- A. $4K \times 1$.
 - B. $4K \times 4$.
 - C. $16K \times 1$.
 - D. $16K \times 4$.
39. The basic storage element in a dynamic RAM is a _____.
40. When a cell is read in a dynamic RAM, the data is destroyed.
- A. True
 - B. False
41. The storage element in a three transistor dynamic cell is the MOS-FET _____.
42. The sense/write amplifier in a one transistor dynamic cell memory is a _____.
43. The process of restoring data after a read operation and periodically to retain data is called _____.
44. In 16K and larger RAMs, the _____ input lines are multiplexed.
45. The largest available dynamic RAM can store _____ bits.
46. The IC used to handle all multiplexing operations for a dynamic RAM is called a _____.

Answers

35. B — Dynamic RAM

36. A — Static

37. B — Dynamic

38. C — $16K \times 1$

39. capacitor

40. A — True. This is called destructive read out.

41. gate capacitance

42. latch

43. refresh

44. address

45. 256K

46. refresh controller

PROGRAMMABLE READ-ONLY MEMORIES

Read only memories, like random access read/write memories, have undergone considerable evolutionary changes in a short period of time. Innovations in both bipolar and MOS technology have resulted not only in larger storage capacity ROMs, but also in the development of a special category of ROM known as the programmable read only memory (PROM). The increased storage capacity and the ability of the user to program the PROM in the field have made ROMs much more useful and acceptable in computer and other digital applications.

ROM operation and application was discussed in detail in Unit 8. These were masked ROMs that are programmed by the manufacturer. In this section, we discuss programmable read only memories. PROMs are user programmable in the field.

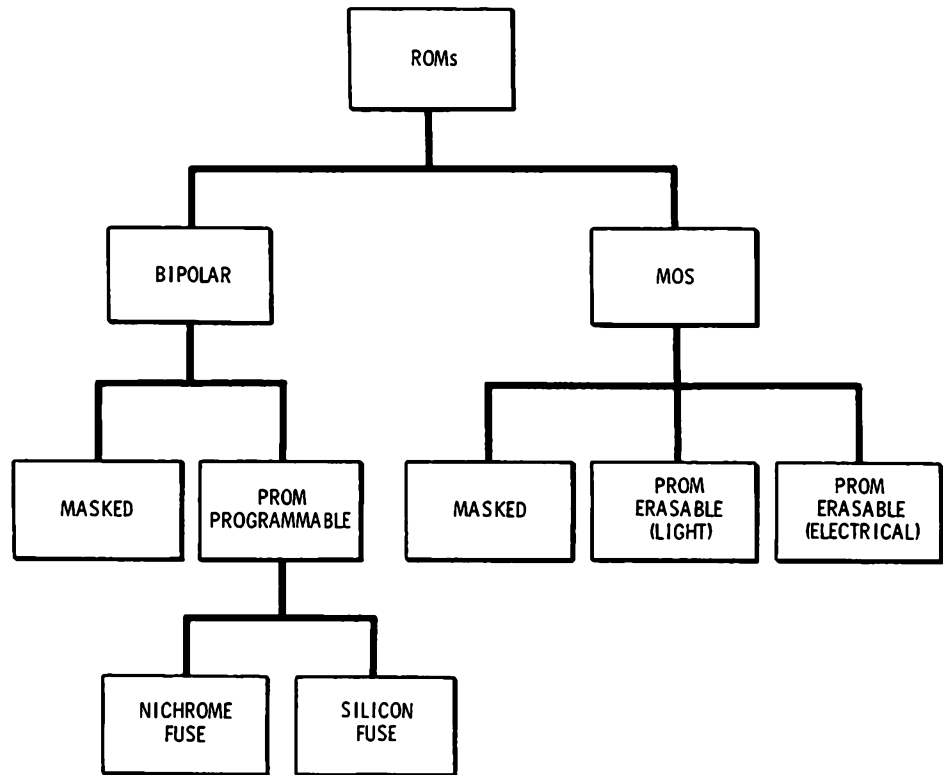


Figure 9-21
Hierarchy of read-only memories.

Figure 9-21 shows the hierarchy of semiconductor ROMs. Note that the two basic categories of ROMs are bipolar and MOS. As you can see, there are two basic types of bipolar ROMs, masked and programmable. Further, Figure 9-21 shows that there are two basic types of fused bipolar PROMs that can be programmed externally after the device has been manufactured.

Now consider the MOS ROMs in Figure 9-21. Again there are masked MOS ROMs. Then there are two categories of PROMs, the light-erasable PROM and the electrically-erasable PROM. These devices can be erased by an external electrical signal or ultraviolet light.

In this section, we will discuss both bipolar and MOS PROMs. In all cases, the main characteristic of these devices is that they can be written into after they are manufactured. In some cases, once the data has been written into it, it is permanent and cannot be changed. In other cases, the PROM can be erased and reprogrammed.

Bipolar PROMs

Recall from your study of ROMs in Unit 8, that a ROM, or any semiconductor memory for that matter, is basically a matrix of semiconductor devices. The matrix is made up of rows and columns of semiconductor devices. The row and column address decoders operate the various memory cells. At the junction of each row and column is a semiconductor device, which can be a diode, a bipolar transistor, or a MOSFET. This device can be programmed so that it either conducts or is cut off. In that way, either a binary 0 or binary 1 is programmed at each row/column juncture. In masked ROMs, the programming takes place by applying or not applying a metalized layer to the device to either cause it to conduct or to be cut off. In a programmable bipolar PROM, the device is usually programmed with a fuse.

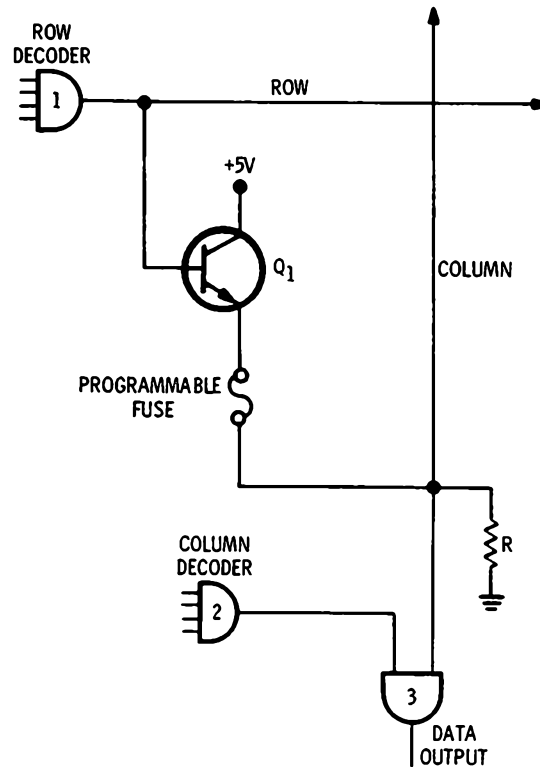


Figure 9-22
Typical fuse programmed bipolar ROM circuit.

Figure 9-22 shows the basic row and column arrangement of the typical ROM. Where each row and column intersect, a bipolar transistor is connected. Note that a fuse is connected in series between the emitter of the transistor and column connection. When the fuse is closed, that transistor is connected, causing it to conduct, which programs a binary 1 in that location. If the fuse is blown or open, the transistor does not conduct. Therefore, a binary 0 is programmed.

When the ROM is manufactured, all of the fuses are good and, therefore, all transistors conduct. In other words, the bipolar PROM comes from the manufacturer with all binary 1's. To program the device, external circuitry is used to "blow" the fuses in the desired location to program binary 0's. The PROM is usually programmed by applying a short duration, high current pulse to the desired bit. This blows or opens the fuse and programs the binary 0.

The circuit in Figure 9-22 operates as follows. To select the bit shown, the inputs to both row and column decoders are binary 1's. Gates 1 and 2 have binary 1 outputs as a result, which enables gate 3. If the fuse is good (closed), gate 1 causes Q1 to conduct. Q1 acts as an emitter follower and applies a binary 1 to gate 3, and the data output is a binary 1.

If the fuse is open, Q1 will not conduct. Resistor R will keep the input to gate 3 at ground or binary 0, and the data output will be binary 0.

NICHROME FUSE

The first bipolar PROM fuses were made of nichrome (an alloy of nickel and chrome), deposited as a very thin film link between the emitter and the column lines of the PROM.

Nichrome was chosen for the fuse material primarily because it was easy to blow with reasonably low currents. However, it has now been abandoned because it is difficult to work with in manufacturing integrated circuits, and exhibits the phenomenon referred to as "growback". Once a nichrome fuse is blown, it does present an open circuit, but the way in which it is blown does not make it totally reliable. After some time, the two nichrome pieces tend to grow back together, thus bridging the open circuit and creating a resistive path. This in effect causes the previously programmed binary 0 to eventually become binary 1 again. While improvements have been made in the nichrome fuse process, for the most part, nichrome is no longer used in modern PROMs.

SILICON FUSE

Another type of fuse used in bipolar PROMs is the silicon fuse. A thin strip of the fuse material (polycrystalline silicon) is deposited on the oxide in the manufacturing process. Its thickness and cross-sectional area is made so that it can be blown with a current of approximately 20 to 30 mA, and there is no "growback" problem with silicon fuses.

TYPICAL BIPOLAR ROMs

Typical bipolar ROMs come in a variety of configurations. Both TTL and ECL PROMs are available, but the TTL devices are the most popular. Typical memory sizes range from 256 bits to 16K bits. Most ROMs are organized as multiple storage locations for 8-bit bytes. The smallest bipolar ROM is a 32×8 , 256 bit PROM. The largest bipolar PROM contains 16K bits, and is organized in $2K \times 8$ configuration. Larger ROM storage capacities can only be obtained with MOS circuitry.

The primary benefit of bipolar PROMs are their high speed. Typical TTL PROM access times are in the 50-to-80 nanosecond range. ECL PROMs are also available with access times of less than 20 nanoseconds.

MOS PROMs

MOS PROMs are typically made so that they can be erased and reprogrammed. Known as erasable programmable memories (EPROMs), these devices can be erased by light or an electrical signal.

LIGHT EPROM

The most popular and widely used PROM is a MOS device that can be erased by exposing it to ultraviolet light. The EPROM (erasable programmable read only memory) offers the high density of MOS technology with the convenience of erasing and reprogramming.

Like most semiconductor memories, the EPROM is organized as a matrix of rows and columns with a MOSFET at each row/column junction. The MOSFET is programmed so that it conducts or is cut off to program binary 1's and 0's. The unique feature of such a PROM is the special architecture of the MOSFET. This device features a special "floating" gate (the gate has no external lead, and is not connected to any external circuitry). This special MOSFET is programmed by charging the gate. If the gate is charged, the MOSFET will conduct; if the gate is not charged, the MOSFET will be cut off. Programming the EPROM then is a process of charging those gates where it is desirable to store a binary 1.

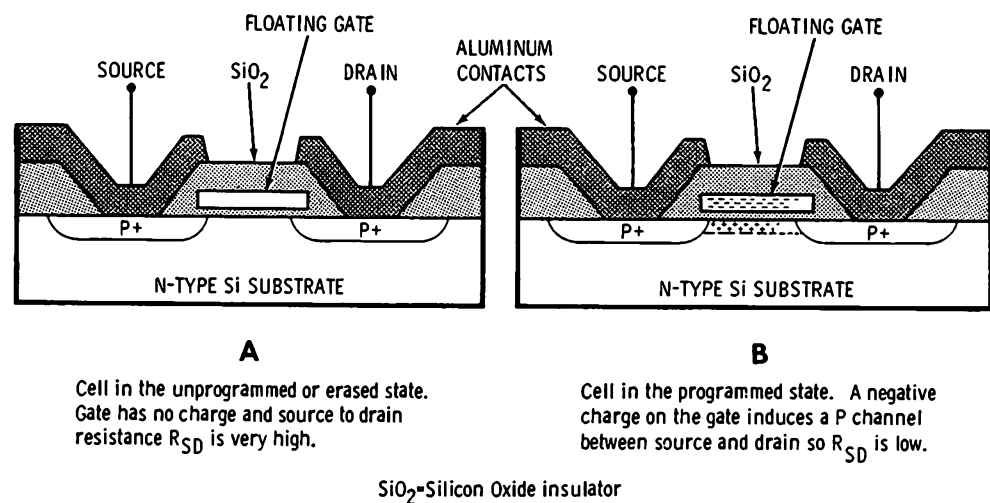


Figure 9-23
Cross sectional view of floating gate MOSFET used in EPROM.

Figure 9-23 shows a cross-sectional view of an EPROM MOSFET. It consists of an N-type substrate into which has been diffused two P-type areas to form the source and the drain. Aluminum connections are attached to the source and drain areas. A silicon oxide layer on the substrate between the source and the drain contains a floating silicon gate that is electrically insulated from the source and drain regions. The result is a P-type MOSFET.

When the MOSFET is not programmed, there is no charge on the gate. As a result, the source-to-drain resistance is extremely high. A binary 0 is said to be programmed into the MOSFET at this time.

To program a binary 1 into the MOSFET, the MOSFET must be made to conduct. Ordinarily to cause a P-channel enhancement mode MOSFET to conduct, a negative gate voltage would be applied. To cause an EPROM MOSFET to conduct, the floating gate must somehow be charged negatively.

To program an EPROM MOSFET, a source-to-drain voltage of approximately 25 to 30 volts is applied between the source and the drain. This voltage is high enough to cause an avalanche breakdown in the P-N junctions existing between the source and the substrate, and the drain and the substrate. Some of the electrons flowing as a result of the avalanche breakdown, penetrate the silicon oxide barrier and collect on the isolated gate. A negative charge, therefore, builds up on the gate. The higher the avalanche voltage and the longer it is connected, the higher the charge becomes. With the gate charged, and the programming voltage removed, the MOSFET is now biased on.

The process of programming the EPROM MOSFET then is one of applying high source-to-drain voltage for a given period of time. With the MOSFET conducting, a binary 1 is programmed. Keep in mind that while a P-channel device was discussed here, N-channel devices can also be manufactured.

Because of the very low leakage of the silicon oxide, the charge programmed on the gate will be retained for a very long time. The leakage is so low in fact, that the gate charge is retained for many years. The programming is, for most practical purposes, permanent.

Once an EPROM is programmed, you can erase it by exposing the matrix of floating gate MOSFETs to ultraviolet light. Most PROMSs are housed in dual in-line packages with the chip centered in the package. Over the chip is a quartz cover that will admit light. To erase the EPROM, ultraviolet light is applied to the chip for a period of 10 to 20 minutes. (No more than about 20 minutes is required to completely erase the PROM.) Ultraviolet light causes the charge on the gates to be dissipated.

Ordinary light, including sunlight, normally contains some small amount of ultraviolet rays. Usually they are inadequate to erase the EPROM. Nevertheless, most EPROMs are protected from ordinary light by a small piece of tape applied over the quartz window on the IC.

The first EPROM, and one of the most popular, was the model 1702A. This device is a 256×8 , 2K bit device. Although programming was slow (it took many minutes to charge the cells adequately enough to ensure reliability) and access times were long, the 1702A was widely used because it could be erased and reprogrammed in the field.

Since then, new EPROMs have been developed. The popular 2700 series is a good example. This is a series of 8K, 16K, 32K, and 64K bit devices. For example, the 2708 is an 8K bit device organized in a $1K \times 8$ configuration, the 2716 is a 16K device organized in a $2K \times 8$ configuration, and the 2732 and 2764 EPROMs have $4K \times 8$ and $8K \times 8$ organizations, respectively. The largest current EPROM can store up to 64K bits.

These N-channel devices operate from a single 5-volt supply. In addition, they have the advantage of being programmed in a relatively short period of time. All of them can be erased with ultraviolet light.

ELECTRICALLY ERASABLE PROM

Another popular form of EPROM is the electrically-erasable PROM, EEPROM, or E²PROM. Instead of using ultraviolet light, you can erase this device by applying appropriate electrical signals to the chip circuitry. This is an important advance in ROM design, as it combines the read and write capability of a RAM but with the nonvolatile characteristics of a ROM.

Typical E²PROMs are available in $1K \times 8$ and $2K \times 8$ organizations. They are made with standard floating-gate MOSFETs. You can erase the contents of the ROM on a byte-by-byte basis by addressing the desired byte and applying a single 21-volt pulse. A chip erase capability is also provided. Again, a 21-volt pulse, applied to the circuit under certain conditions, erases the entire contents of the ROM.

Any byte or the entire chip can be erased or written into in 10 milliseconds. Although 10 milliseconds is a long write time compared to the times normally dealt with in typical digital systems, the E²PROM is extremely valuable. It permits data to be stored in a ROM and then read later under the control of a microcomputer or other portion of the digital system.

In all systems containing standard read/write memories, removing the power causes the data in the memory to be lost. When an E²PROM is used, any data stored in the ROM will be retained when power is removed. Very high-speed read-access times of several hundred nanoseconds and the benefit of nonvolatility makes the E²PROM one of the most desirable semiconductor memory devices available.

PROM Programmers

Both bipolar and MOS PROMs are usually programmed by a special piece of equipment known as a PROM programmer, also known as a PROM “burner” or PROM “blaster.” The PROM programmer contains all of the circuitry for addressing a PROM and storing the desired bits, as well as an address counter that is incremented to sequentially address the ROM words. Also included are the circuits for generating the high voltage programming pulses, which either blow the internal fuses or charge the isolated gate MOSFETs. Most modern PROM programmers can be used to program a wide variety of PROMs. Through the use of “personality modules”, the PROM programmer can generate the necessary programming pulses for the most popular types of bipolar and MOS ROMs. These “personality modules” are often plug-in devices.

PROM programmers usually contain a hexadecimal keyboard and 7-segment LED displays. The keyboard allows an operator to enter a desired address and then the desired data. The operator can initiate the programming operation by pressing a “store” button. You can program PROMs manually by sequentially entering the addresses and then the data in hexadecimal form. The LED display shows the address and data programmed. With these manual controls, you can enter any address, and program data. You can also view the contents of each address to be sure that the data was properly programmed.

Another feature of most PROM programmers is a copy capability. This allows the contents of an existing PROM or ROM to be copied into a new PROM. The PROM programmer contains sockets for both the ICs. Automatic circuitry in the PROM programmer sequentially steps through the addresses of both the ROM to be copied and the new PROM. The data is sequentially transferred from one ROM to the other.

As PROMs continued to get larger in size, it became evident that manual programming was impractical. It would be a tedious, time-consuming, and error-prone process to manually enter addresses and data to program a PROM with, for example, 8K bytes. As a result, most PROM programming today is done automatically.

The addresses and data are generally supplied by an external microcomputer interfaced to the PROM programmer. Usually, the content of a PROM is a series of instructions and data words used by a microcomputer. This information is typically generated in a microcomputer and stored in its memory (RAM). That data is then transferred from the microcomputer to the PROM programmer where it is entered into the PROM.

Self Test Review

47. ROMs into which data can be written in the field after manufacture are said to be _____.
48. The two main categories of PROM are _____ and _____.
49. PROMs are nonvolatile.
- A. True
B. False
50. Bipolar PROM storage cells contain a _____ used for programming.
51. The largest bipolar PROM can store up to _____ bits.
52. The largest MOS EPROM can store up to _____ bits.
53. Bipolar PROMs are available with both _____ and _____ circuits.
54. MOS PROMs are faster than bipolar PROMs.
- A. True
B. False
55. The two types of fuses used in bipolar PROMs are _____ and _____.
56. Growback is sometimes a problem in _____ type fuses.
57. Most MOS PROMs can be _____.
58. The two basic methods of clearing all data from a MOS PROM are _____ and _____.
59. The main storage element in a EPROM is a _____ MOSFET.

60. EPROMs are erased by applying _____ light.
61. EEPROMs are erased by an _____ signal.
62. The instrument used to store data in a PROM is called a _____.

Answers

- 47. programmable
- 48. bipolar, MOS
- 49. A — True
- 50. fuse
- 51. 16K
- 52. 64K
- 53. TTL, ECL
- 54. B — False Bipolars are faster than MOS.
- 55. nichrome, silicon
- 56. nichrome
- 57. erased
- 58. light and electrical
- 59. floating gate
- 60. ultraviolet
- 61. electrical
- 62. PROM programmer

EXPERIMENT 23

Semiconductor Memories

OBJECTIVES: *To demonstrate the operation of a static RAM semiconductor memory IC.*

Introduction

In this experiment, you will gain practical experience in using semiconductor memory ICs. Specifically, you will demonstrate the operation and application of a typical static RAM. The device you will be using will be the popular 2114 1K \times 4 static RAM. The operation of this device was described earlier in the unit.

You will apply addresses to the memory chip and then perform both read and write operations, and show the function of both the chip enable and read/write control signals. Further, you will demonstrate the concept of volatility.

This experiment, as with many others you have implemented previously, is made up of a number of integrated circuits. Be careful in wiring the circuit to avoid interconnection errors. This is the most common cause for a malfunctioning circuit. Be sure that all ICs are connected to +5 volts and ground. Refer to Figure 9-11 for pin-out details on the 2114.

Materials Required:

Heathkit Digital Design Experimenter ET-3200

1—2114 RAM IC (443-764)

1—74193 IC (443-815)

1—14495-1 IC (443-1802)

1—7-segment LED display

1—1 k Ω resistor

Procedure

1. Construct the circuit shown in Figure 9-24. The 74193 4-bit binary counter will be used to supply addresses to the 2114 RAM. The decoder driver and 7-segment LED display will provide a convenient readout of the address in hexadecimal notation. Logic switch A will be used to step the counter and thus change the RAM address. Logic switch B will be used to control the read/write function of the 2114 RAM chip. The four LED displays on the Trainer will be used to read the data stored in the RAM.
2. Examine the circuit shown in Figure 9-24. Note that the 2114 has a 10-bit address input A0 – A9. Assume that the 74193 counter contains the binary number 1111. The address applied to the 2114 RAM, therefore, is _____.

The LED display will read out the character _____ at this time.

3. Apply power to the circuit. Increment logic switch A so that the counter is stepped until the 7-segment display reads 0. Then step the counter through its 16 states as indicated by the 7-segment display showing the digits 0 through 9 and A through F. For each applied address, note the reading on the 4-bit LED display. Record the display contents in Table I Page 9-79.

How do you account for the contents of the RAM at this time?

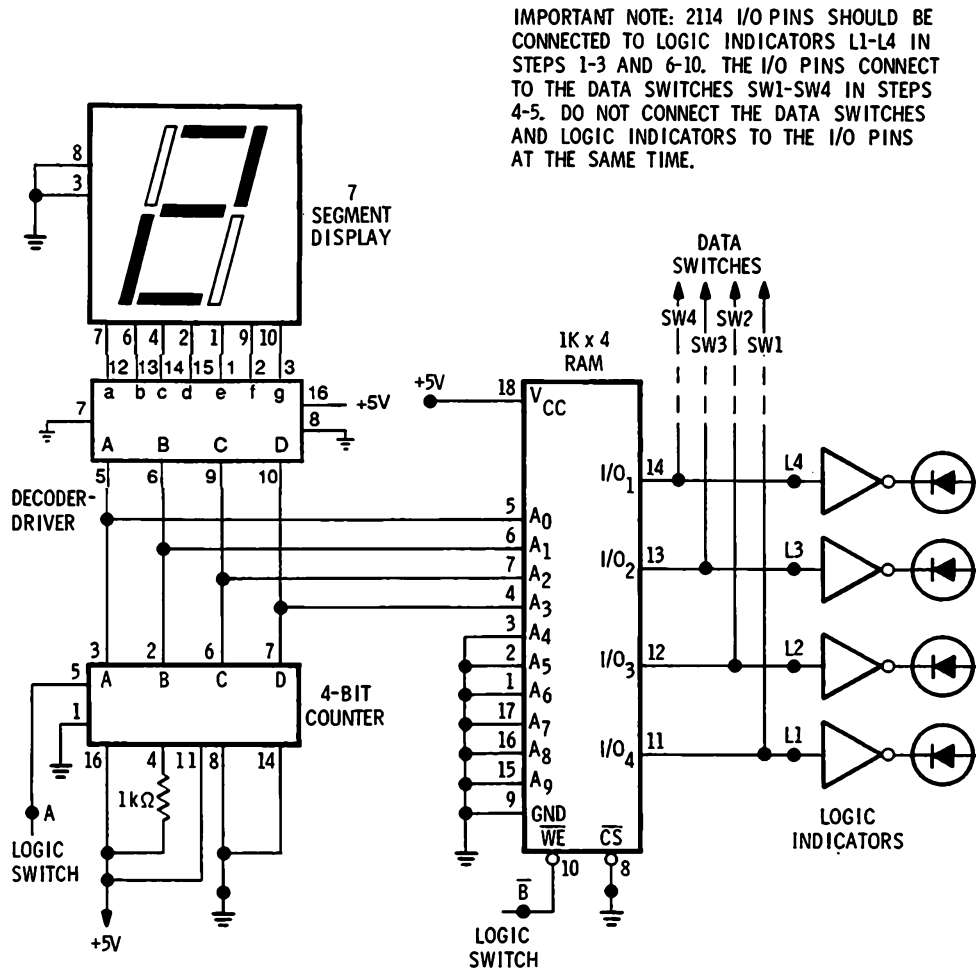


Figure 9-24
Experimental circuit for Steps 1 through 10.

Discussion for Steps 1 – 3

In the circuit you constructed, you are using a 4-bit binary counter to apply a 4-bit address to the four lower order bits of the RAM address input. The six most significant bits are binary 0's because pins 3, 2, 1, 17, 16, and 15 are connected to ground. Therefore, the addresses being applied to the RAM are 0000000000 through 0000001111. With 1111 applied to the 2114, the 7-segment display will read a hexadecimal F. For this experiment, we will only be using the lower 16 locations of the RAM chip because of this limited addressing capability. Keep in mind that the 2114 RAM is capable of storing 1024 (1K) 4-bit words.

When you apply power to the circuit, the 2114 RAM is fully functional. However, applying power to a RAM chip will cause random data to appear in the device. As the power turns on, the storage cells in the RAM can come up in either a set or reset state, so the content of the memory on power up can be virtually anything.

As you incremented the lower order address bits, you read the contents of the first 16 memory locations. Logic switch B connected to pin 10 applies a binary 1 level to the read/write or write enable line (\overline{WE}), which tells the RAM chip to perform the read function. The chip select (\overline{CS}) input is grounded, therefore, the RAM is enabled. The output of the RAM is displayed on LED indicators L1 through L4. The data you recorded in Table I should have shown a random selection of 4-bit binary numbers. The data is meaningless.

Procedure (continued)

4. Turn off the power to the circuit. Modify the circuit to connect the 2114 I/O lines to the data switches **instead** of the logic indicators. See Figure 9-14. Data switches SW1 through SW4 will be used as a binary word source for the RAM. You will set these switches to various states and store them in sequential memory locations.
5. Increment logic switch A until the 7-segment LED display reads 0. Then set the data switches to 0000. Depress logic switch B momentarily and then release it. Next, increment logic switch A to the next address, which will be 1. Then set the four data switches to the binary equivalent of the address being displayed on the 7-segment display, or 0001. Again momentarily increment logic switch B. Continue to repeat this process where you increment the counter with logic switch A, set the data switches to the binary equivalent of the address shown on the 7-segment display and then store the data by momentarily depressing logic switch B. Once you have stored all 16 states in the RAM, this step is complete. Set all data switches to 0000.

Discussion for Steps 4 and 5

In these steps, you rewired your circuit to use the data switches as a switch register to supply binary numbers to the RAM. You then incremented the RAM address and stored in each sequential location the binary equivalent of the hexadecimal address being displayed. Each time you depress logic switch B, you switch the read/write (\overline{WE}) control line from read to write, thereby storing the data in the memory location.

Procedure (Continued)

6. Rewire the circuit, replacing the data switches with the logic indicators. **Do not turn off the power to make this wiring change.** The wiring changes are minor, and all you are doing is resubstituting the LED display for the logic switches. Be careful as you make this change so that adjacent pins or wires do not touch. Be careful so you do not damage the circuit.
7. Disconnect the R/W input jumper from the logic switch B output connector. Reconnect the R/W input jumper to the logic switch \overline{B} output connector. Using logic switch A, step the address counter through its 16 states. At each address, note the content of that RAM location. Record the reading of the LEDs in Table II.
8. Remove the wire connecting the chip select (\overline{CS}) input on pin 8 from its ground connection. Connect pin 8 of the 2114 RAM to a convenient +5-volt terminal. Now increment logic switch A through the 16 memory locations and observe the states of LEDs L1 through L4.

With the chip select input at +5 volts, the output LEDs display the same states for each input address as when the chip select is connected to ground. True or false?

9. Reconnect chip select pin 8 on the 2114 to ground. Again step through the 16 input addresses with logic switch A. Again note the LED display for each memory address.
10. Using the power switch on the ET-3200 Experimenter, turn off the power for ten seconds and turn it back on. Using logic switch A, step through the 16 addresses from 0 through F and record the RAM contents in Table III. Compare the contents in Tables II and III.

Account for any differences between Tables II and III.

TABLE I

ADR	L1	L2	L3	L4
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
A				
b				
C				
d				
E				
F				

TABLE II

ADR	L1	L2	L3	L4
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
A				
b				
C				
d				
E				
F				

TABLE III

ADR	L1	L2	L3	L4
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
A				
b				
C				
d				
E				
F				

Discussion for Steps 6 – 10

In step 6, you sequenced through the input addresses and observed the contents of the RAM stored there in the previous steps. You should have found that at each memory location, the contents of the RAM was the 4-bit binary number corresponding to the hex address being displayed on the 7-segment readout.

Next, you connected pin 8 of the 2114 to +5 volts. Pin 8 is the chip select input. As long as pin 8 is held at ground, the chip is enabled. This means that all of the circuitry on the chip is fully operational and the device is capable of performing both read and write operations. When pin 8 is connected to binary 1 or +5 volts, the RAM chip is completely disabled. The data stored in the memory, of course, is retained. However, it is not possible to perform either read or write operations with the chip disabled. Recall that the 2114 has 3-state output drivers. When the chip select input is high, the 3-state drivers are in their high impedance state. The four output lines connected to the LED drivers on the ET-3200 Experimenter, therefore, are effectively open. At this time, displays L1 through L4 should be binary 0s for all input addresses. Returning pin 8 to ground thereby enables the memory. By sequencing through the addresses, you should find that the data previously stored there can again be read out.

In the final step, you momentarily switched off the power. Because of the volatility of the 2114 semiconductor memory, all of the data previously stored there was lost. You discovered this when you reapplied power and stepped through the 16 addresses. You should have found random binary numbers in the RAM at this time rather than the numbers 0000 through 1111 stored there in previous steps. This clearly demonstrates that even a momentary loss of power erases all data stored in a semiconductor memory chip.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this Unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and work every problem first before checking the answer.

1. Which of the following non-semiconductor devices are still used in digital memories?
 - A. Flip-flops.
 - B. Relays.
 - C. Vacuum tubes.
 - D. Magnetic cores.

2. Storing data into a memory is called:
 - A. Writing.
 - B. Stashing.
 - C. Memorizing.
 - D. Remembering.

3. Retrieving data from a memory is called:
 - A. Getting.
 - B. Reading.
 - C. Fetching.
 - D. Accessing.

4. A ROM is also a:
 - A. Serial access memory.
 - B. Read/write memory.
 - C. Random access memory.
 - D. Volatile memory.

5. The basic storage circuit of a sequential access digital memory is a:
 - A. Counter.
 - B. Shift register.
 - C. Capacitor.
 - D. One shot.

6. The number given to a memory location for reference purposes is called the:
- A. Mantissa.
 - B. Code.
 - C. Instruction.
 - D. Address.
7. To sequentially store 4096 7-bit ASCII words, a shift register must have how many bits?
- A. 16,384.
 - B. 28,672.
 - C. 32,768.
 - D. 65,536.
8. The loss of data from a power failure in a semiconductor memory is called:
- A. Volatility.
 - B. Nonvolatility.
 - C. Erasure.
 - D. Nonpermanence.
9. The interval required to address and read out a memory word is called:
- A. Access time.
 - B. Pulse duration.
 - C. Settling time.
 - D. Propagation delay.
10. In memory terminology, K means:
- A. 1,000.
 - B. 1,024.
 - C. 10,000.
 - D. 100,000.

11. Which configuration below refers to a memory that can store 32,768 bytes?
- A. $8K \times 32$.
 - B. $32K \times 4$.
 - C. $32K \times 8$.
 - D. $32,768 \times 4$.
12. How many $16K \times 1$ memory ICs does it take to form a $128K \times 8$ memory?
- A. 16.
 - B. 32.
 - C. 64.
 - D. 128.
13. The standard designation for a read/write memory is:
- A. RAM.
 - B. ROM.
 - C. RWM.
 - D. RMM.
14. Which of the following are **not** advantages of a dynamic RAM over a static RAM?
- A. Easier to use.
 - B. Lower power consumption.
 - C. Smaller size, greater chip density.
 - D. Faster.
15. More bits may be stored in a given area with dynamic cells because:
- A. Dynamic cells are larger.
 - B. Dynamic cells are smaller.
 - C. Dynamic cells consume less power.
 - D. Dynamic cells are faster.
16. The access time of a static RAM is predominately:
- A. Charge/discharge time.
 - B. Propagation delay.
 - C. Speed of external circuits.
 - D. Storage element characteristics.

17. Which of the following is **not** a typical characteristic of a bipolar RAM?
- A. High power consumption.
 - B. High speed.
 - C. Nonvolatility.
 - D. Small storage capacity.
18. Typical access times for a MOS dynamic RAM are:
- A. 5–20 ns.
 - B. 20–100 ns.
 - C. 100–500 ns.
 - D. 500–900 ns.
19. A $64\text{K} \times 8$ memory can store how many bytes?
- A. 32K.
 - B. 64K.
 - C. 128K.
 - D. 256K.
20. Which organization is most typical for a PROM?
- A. $4\text{K} \times 1$.
 - B. $4\text{K} \times 4$.
 - C. $4\text{K} \times 8$.
 - D. $4\text{K} \times 16$.
21. The flip-flop is the main storage element in which kind of memory?
- A. Static.
 - B. Dynamic.
22. The main storage element in a dynamic memory is a:
- A. Flip-flop.
 - B. Capacitor.
 - C. One shot.
 - D. MOSFET.

23. Lowest power consumption is obtained with which type of memory?

- A. ECL.
- B. TTL.
- C. MOS.

24. The memory IC input that “turns on” the circuitry in preparation for a read or write operation is called the:

- A. Address.
- B. Read/write.
- C. Chip enable.

25. A memory IC with a 128×128 storage matrix can retain how many data bits?

- A. 128.
- B. 16K.
- C. 64K.
- D. 128K.

26. Another name for the read/write control line is:

- A. Write enable.
- B. Chip select.
- C. Address.

27. In multi-IC memories, extra decoders enable the selected chips through which IC inputs?

- A. Address.
- B. Read/write.
- C. Data input.
- D. Chip select.

28. Restoring the charge on a dynamic storage cell is called:

- A. Refresh.
- B. Store.
- C. Write.
- D. Retain.

29. The refresh operation in a typical dynamic RAM takes place every:
- A. 100ns.
 - B. 2 μ s.
 - C. 10 μ s.
 - D. 2ms.
30. The refresh operation is usually performed by a special IC called a:
- A. Binary counter.
 - B. Dynamic RAM controller.
 - C. Shift register.
 - D. Multiplexer.
31. ROMs into which data can be written after the ROM is made are called:
- A. RAMs.
 - B. ROMs.
 - C. PROMs.
 - D. RWMs.
32. Bipolar PROMs are programmed by:
- A. Biasing a bipolar transistor.
 - B. Blowing a fuse.
 - C. Masking a bipolar transistor.
 - D. Charging a floating gate.

33. The main storage element in a MOS PROM is a:
- A. Fuse.
 - B. MOSFET.
 - C. Floating gate MOSFET.
 - D. Bipolar transistor.
34. EPROMs can be erased by:
- A. Blowing a fuse.
 - B. Applying a 21-volt pulse.
 - C. Turning off the power.
 - D. Applying ultraviolet light.
35. The device used to store data in a PROM is called a/an:
- A. PROM programmer.
 - B. Microcomputer.
 - C. Paper tape reader.
 - D. Ultraviolet light source.

EXAMINATION ANSWERS

1. D — Magnetic cores.
2. A — Writing.
3. B — Reading.
4. C — Random access memory.
5. B — Shift register.
6. D — Address.
7. B — $28,672 (7 \times 4096 = 28,672)$.
8. A — Volatility.
9. A — Access time.
10. B — 1024.
11. C — $32K \times 8$ (byte = 8 bits).
12. C — $64 \quad 128K \times 8 = 1024K$.
 $1024/16K = 64$.
13. A — RAM.
14. A — Easier to use.
D — Faster.
15. B — Dynamic cells are smaller.
16. B — Propagation delay.
17. C — Nonvolatility.
18. C — 100 – 500ns.
19. B — 64K.
20. C — $4K \times 8$.

21. A — Static.
22. B — Capacitor.
23. C — MOS.
24. C — Chip enable or chip select.
25. B — 16K.
26. A — Write enable.
27. D — Chip select.
28. A — Refresh.
29. D — 2ms.
30. B — Dynamic RAM controller.
31. C — PROMs.
32. B — Blowing a fuse.
33. C — Floating gate MOSFET.
34. D — Applying ultraviolet light.
35. A — PROM programmer.

Unit 10

DATA CONVERSION

CONTENTS

Introduction.....	10-3
Unit Objectives.....	10-4
Unit Activity Guide.....	10-5
Purpose of Data Conversion.....	10-6
Digital-to-Analog Conversion.....	10-8
Experiment 24 — Digital-to-Analog Conversion.....	10-40
Analog-to Digital Conversion.....	10-51
Experiment 25 — Analog-to-Digital Conversion.....	10-88
Unit Examination.....	10-98
Examination Answers.....	10-103

INTRODUCTION

Electronic data exists in two basic forms, analog and digital. As you recall, analog data is continuous in nature. Typical analog signals are voltages and current that vary smoothly or continuously in amplitude and shape. Digital data on the other hand consists of binary numbers and codes that represent quantities. Both types of data are widely used in electronics. Frequently, it is necessary to convert data from one form to another. For example, it may be desirable to convert an analog signal into digital numbers that can be processed by a computer. Or similarly, it may be desirable to convert the numerical or tabular output data from a computer into an analog signal. Special circuits known as digital-to-analog (D-to-A) converters and analog-to-digital (A-to-D) converters are designed for this purpose.

In this unit you are going to study various data conversion methods. You will learn several different types of D-to-A and A-to-D conversion methods. We will also discuss support elements for data conversion, including sample and hold circuits and multiplexers.

Review the "Unit Objectives" now. Then begin immediately with the first item in the "Unit Activity Guide."

UNIT OBJECTIVES

When you complete this unit, you will have the knowledge and skills indicated below. You will be able to:

1. Name two reasons why it is desirable to convert from one form of data to another.
2. Explain two basic methods of digital-to-analog conversion.
3. Name and explain the error sources in digital-to-analog conversion.
4. List four different types of analog-to-digital conversion.
5. Explain how sampling rate affects data conversion accuracy.
6. Name and explain the error factors in analog-to-digital conversion.
7. Define the term "time division multiplexing" and explain how an analog multiplexer works.
8. Define and explain the operation of a sample/hold circuit.

UNIT ACTIVITY GUIDE

**Completion
Time**

- ☐ Read "The Purpose of Data Conversion" and "Digital-to-Analog Conversion."

- ☐ Complete Self-Test Review Questions 1 - 15.

- ☐ Perform Experiment 24.

- ☐ Read "Analog-to-Digital Conversion."

- ☐ Complete Self-Test Review Questions 16 - 36.

- ☐ Perform Experiment 25.

- ☐ Complete the Unit Examination.

- ☐ Review the Examination Answers.

PURPOSE OF DATA CONVERSION

In the early days of electronics, virtually all electronic devices and systems were analog in nature. However, with the development of pulse and digital circuits, the designer and the user of electronic equipment soon began to have a choice between analog and digital techniques. In many cases, designers found that specific applications could be optimized or problems solved by combining both analog and digital techniques. When analog and digital techniques are combined within the same piece of equipment, it is generally necessary to convert data from one form to another. Data conversion circuits known as digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) were developed for this purpose.

The single most important development that has increased the need for data conversion hardware is the development and widespread use of microprocessors and digital computers. Today computers are a part of almost any piece of electronic equipment or system. Even the smallest test instruments often contain a computer in the form of a microprocessor.

Microprocessors are widely used because they replace a lot of discrete logic and perform a wide variety of operations under program control. Because of their incredible data processing capacity, microprocessors permit equipment to be more sophisticated, to perform a wider variety of operations, and to implement functions that before were impossible or impractical. Low cost microprocessors bring the power of the digital computer to even the smallest of electronic applications.

Since virtually all electronic instruments and systems today use some form of digital processing, A-to-D and D-to-A conversion techniques are necessary in any applications where analog signals are to be used.

Consider an industrial control system employing a digital computer. Most industrial control systems use a variety of transducers or sensors that monitor analog variables. These variables include temperature, pressure, vibration, liquid level, light intensity, and a variety of other analog information. The transducers convert these analog variables into analog signals. A-to-D converters are used to change the analog data into binary numbers and codes that can be dealt with by the control computer. The computer can store the information in memory or process it.

Most control systems also feature analog outputs. The control computer may wish to generate analog output signals that are used to control servo systems, analog plotters, pen recorders, motors, and other analog devices. To do this, D-to-A converters are used to change the digital data from the computer into the desired analog outputs.

Another modern day application is voice synthesis, which is the process of generating artificial speech with a digital computer. The digital data that forms the speech is passed through a D-to-A converter to form the analog voice output.

In a speech recognition system, the analog voice signal is passed through a microphone to an analog-to-digital converter. The voice is changed into binary numbers and codes which can be recognized and processed by a digital computer.

A-to-D and D-to-A conversion techniques are also used in test instruments. A digital multimeter, for example, is nothing more than an A-to-D converter with a decimal output display. D-to-A converters are widely used in CRT display systems for changing digital data into an analog graphical output that can be displayed on the CRT.

Today, digital techniques are low in cost, sophisticated, and incredibly powerful. In addition to being able to store data very efficiently in high density semiconductor memories, and in digital form on floppy disks and cassette tapes, microprocessors can be used to manipulate the data in a variety of ways. It makes sense, therefore, in most electronic applications to use some form of digital processing. Yet on the other hand, many inputs and outputs that we wish to deal with are in analog form. Data conversion techniques allow us to accept the normal analog inputs and outputs and utilize them most effectively with electronic equipment which is digital in nature.

DIGITAL-TO-ANALOG CONVERSION

A digital-to-analog converter (DAC) is an electronic circuit that changes digital signals into an analog signal. In other words, the DAC converts binary signals or codes into an analog voltage or signal. Figure 10-1 shows a simplified illustration of a DAC.

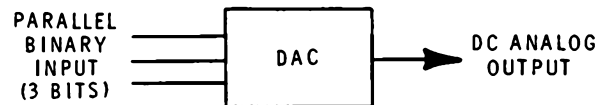


Figure 10-1

Basic concept of a
digital-to-analog converter.

The input to a DAC is usually a parallel binary number. The output is a DC analog voltage that is proportional to the value of the binary input. For example, assume that a DAC accepts a three-bit binary input word. This means that the eight possible input codes from 000 through 111 can be applied to the DAC input. The output of the DAC will be an analog voltage that is proportional to the binary input value. The DAC output may simply be an analog voltage whose DC value equals the binary input equivalent. If the binary input is 111, the analog output voltage will be its decimal equivalent, or 7 volts. In the same way, an input of 100 would produce a 4 volt output, while a 000 input would produce a zero voltage output.

To further illustrate this concept, assume the three-input DAC is connected to the outputs of a three-bit binary counter. See Figure 10-2. As the counter is incremented, it steps from 000 through 111, and then recycles back to 000 and repeats.

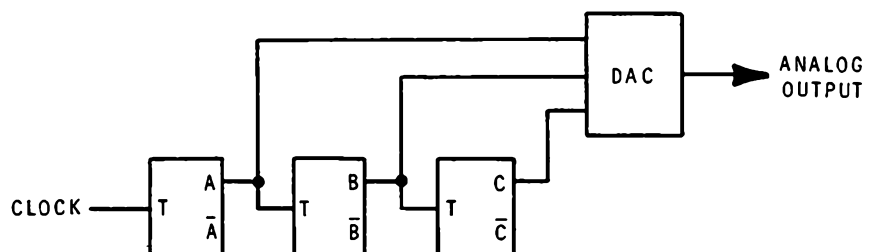


Figure 10-2

3-bit counter driving a DAC.

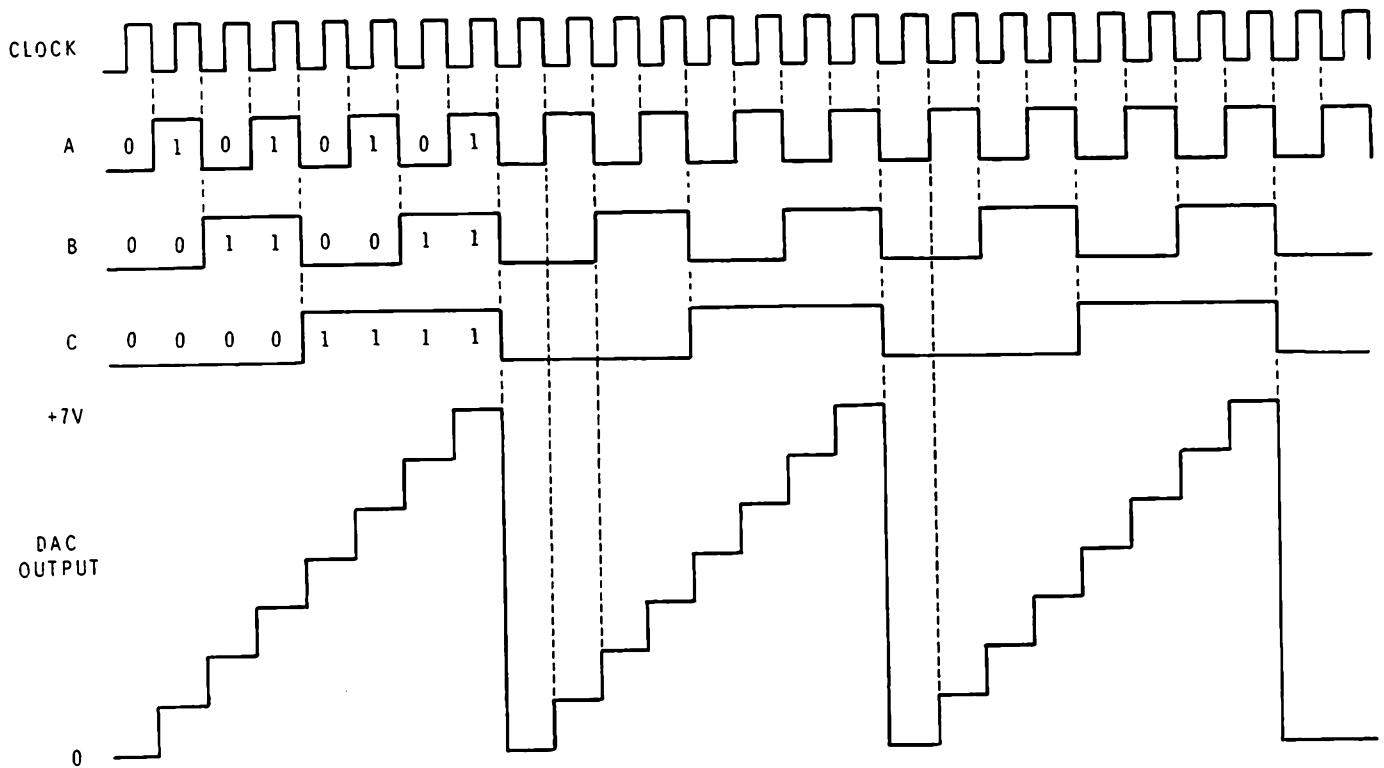


Figure 10-3

Waveforms of binary counter
and DAC output.

Figure 10-3 shows the clock signal stepping the counter, the three binary outputs applied to the DAC, and the resulting DAC output voltage. Note that, as the counter is stepped, the DAC output voltage rises in a linear step fashion. When the maximum value of 111 is applied to the DAC, the maximum output voltage of 7 is produced. On the next input pulse, the counter recycles to 000 and the pattern repeats. Note that the DAC output is a stair-step ramp voltage. It is not truly a smooth continuously varying voltage as a real analog signal would be. However, the waveform very closely approximates a continuously varying linear ramp or sawtooth voltage. This particular analog output waveform varies in one volt increments. In most DACs, the output voltage step can be specified. Further, if more digital input bits are used, there can be many more smaller voltage steps, and the resulting DAC output will even more closely resemble a smooth continuously varying output sawtooth.

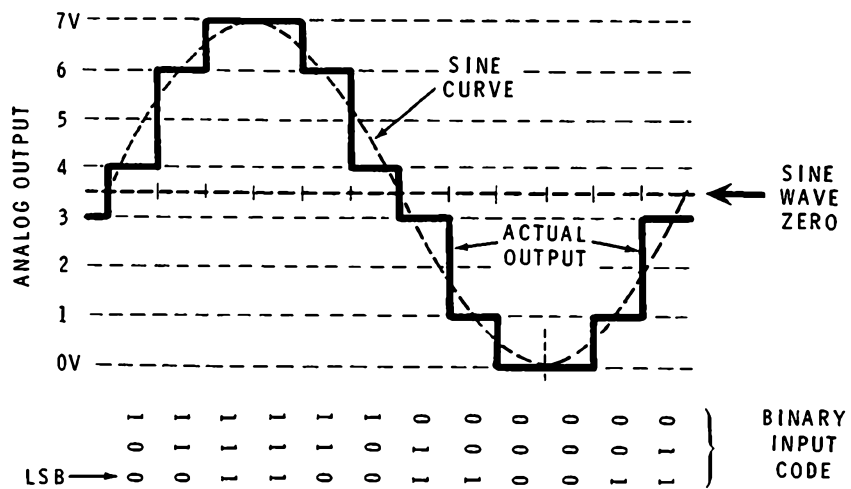


Figure 10-4

Stepped approximation to a sine wave.

While the digital input in this illustration is supplied by a binary counter, thus producing a stair-step ramp output voltage, keep in mind that any sequence of binary inputs may be applied to a DAC. Up-down counters, shift registers, and ROMs are other common sources of input data for a DAC. An excellent approximation to virtually any analog signal can be generated. A fixed binary number input will generate a constant DC output voltage, while a rapidly changing stream of binary numbers will generate a complex analog output waveform. For example, if the sequence of binary numbers in Figure 10-4 is applied to a DAC, the DAC will generate a close approximation to a sine wave. The speed with which the binary input numbers are changed will determine the output frequency of the sine wave.

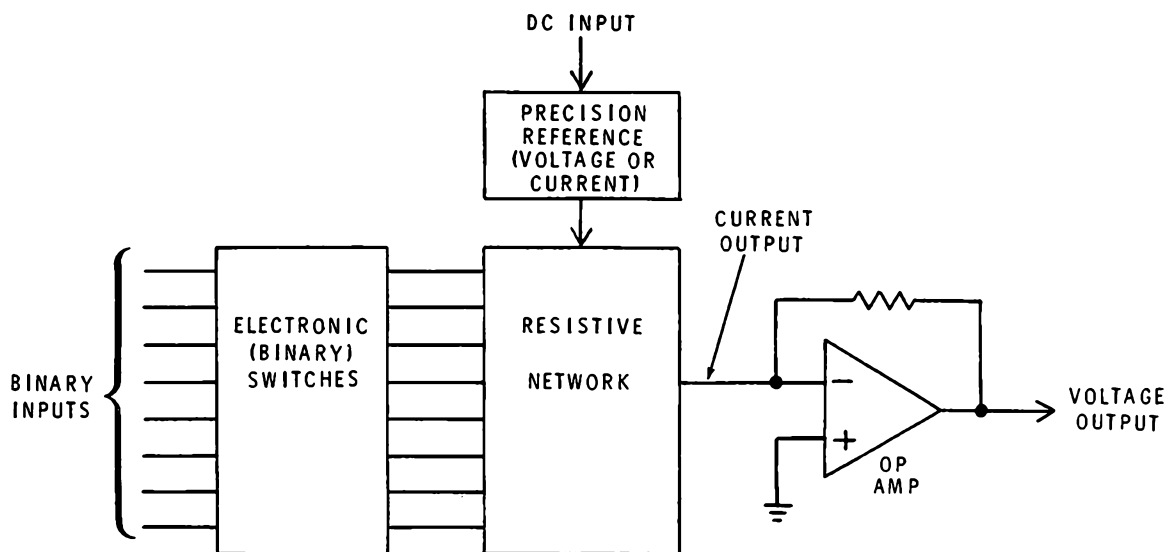


Figure 10-5

Block diagram of a typical DAC.

A more detailed block diagram of a typical DAC is shown in Figure 10-5. It consists of three major parts: a resistive network, a reference source, and binary switches. The digital input word is applied to the binary switches, which control the application of a precision voltage or current reference source to the resistive network. It is the resistive network that actually performs the digital-to-analog conversion. The output of the resistive network is a current that is proportional to the binary input value. The value of the reference source determines the absolute current output.

Normally a current output is not as useful as a voltage signal in most applications. Therefore, a standard operational amplifier is used to change the output current into an output voltage, as shown in Figure 10-5. All DACs use this basic configuration. However, there are wide variations in the type of resistive networks, binary switches, and reference sources used, as you will see.

Binary Weighted Resistor DAC

The simplest form of D-to-A converter is the weighted resistor DAC shown in Figure 10-6. It has all of the basic elements of a DAC as previously discussed. The binary input, in this case a four-bit number, controls binary switches that connect the reference voltage to a resistive network which, in turn, generates a proportional current. The op amp converts the current into the output voltage. The resistors are weighted in a binary fashion where the value of each resistor is twice the value of the next.

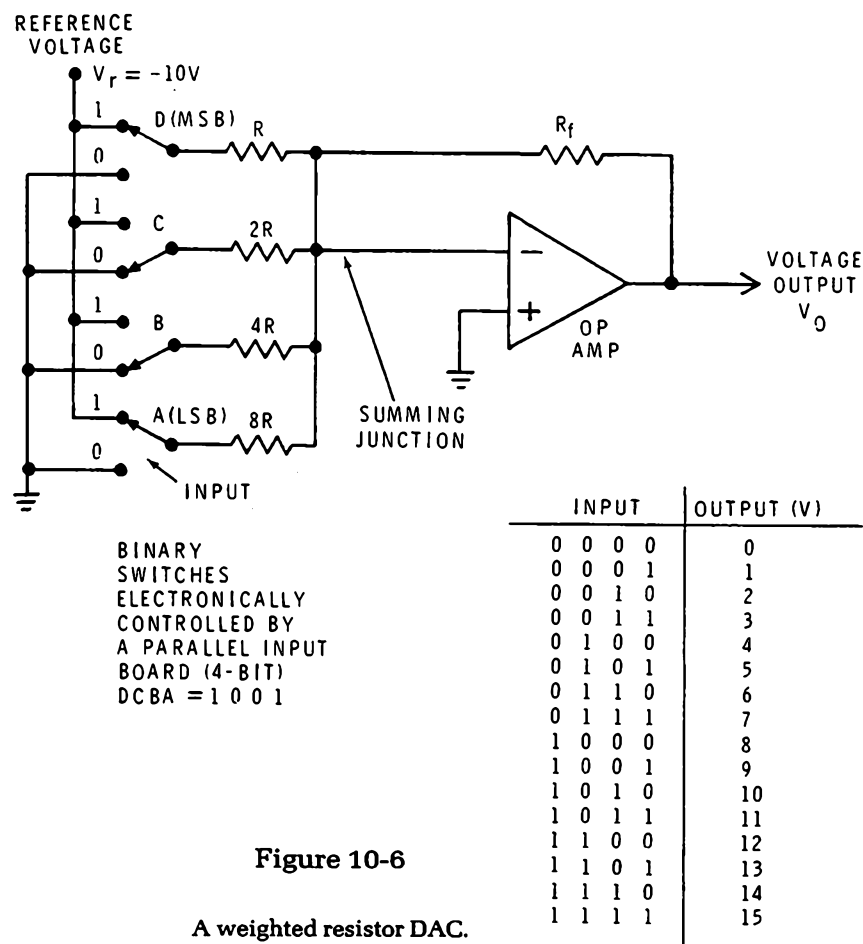


Figure 10-6

A weighted resistor DAC.

A closer look at the circuit in Figure 10-6 shows that it is simply an operational amplifier summer. The voltages applied to the four input resistors are multiplied by a gain factor equal to the ratio of the feedback resistor (R_f) to the value of the input resistor. All of the inputs are then summed to produce the final output voltage. The op amp is connected in the inverting configuration and will generate an output voltage whose polarity is the opposite of its input, in this case, the reference voltage. The formula below expresses the output of the circuit in terms of the input voltages, as well as the input and feedback resistor values.

$$V_o = -V_r(R_f/R + R_f/2R + R_f/4R + R_f/8R)$$

Assuming a -10 volt reference, R is $10k\Omega$, and R_f is $8k\Omega$, you can compute the analog output voltage for any given binary input value. For example, assume that the binary input is 1001. This means that switches A and D are set to the binary 1 position while switches B and C are at the binary 0 positions. Since the input resistors $2R$ and $4R$ are connected to 0 volts, they will contribute nothing to the output. Therefore, the output voltage with the inputs shown is as indicated below.

$$V_o = -(-10)(8k/10k + 8k/80k)$$

$$V_o = 10(.8 + .1) = 9 \text{ volts}$$

As you can see, this is a $+9$ volt output. Because of the binary weighting of the input resistors, the output voltage will be the analog equivalent of the binary input, assuming the proper reference voltage level is used. For other values of reference voltage, the output will not be exactly equivalent to the binary input value; however, it will be proportional.

Figure 10-6 shows a complete summary of the binary inputs and outputs for all 16 possible binary input values. As you can see, the output voltage changes in 1 volt increments. By changing the value of the feedback resistor, different voltage increments can be used. For example, by making the feedback resistor 800 instead of 8k, the output voltage will change in .1 volt steps rather than in 1 volt steps.

In this example, we have shown mechanical switches as being used to connect or disconnect the reference voltage to or from the resistor network. This helps simplify the explanation. In practice, mechanical switches are not used. The idea is to have fast electronic switches controlled by digital circuits so that the DAC can operate at high speeds. Typical electronic switches use bipolar transistors and FETs.

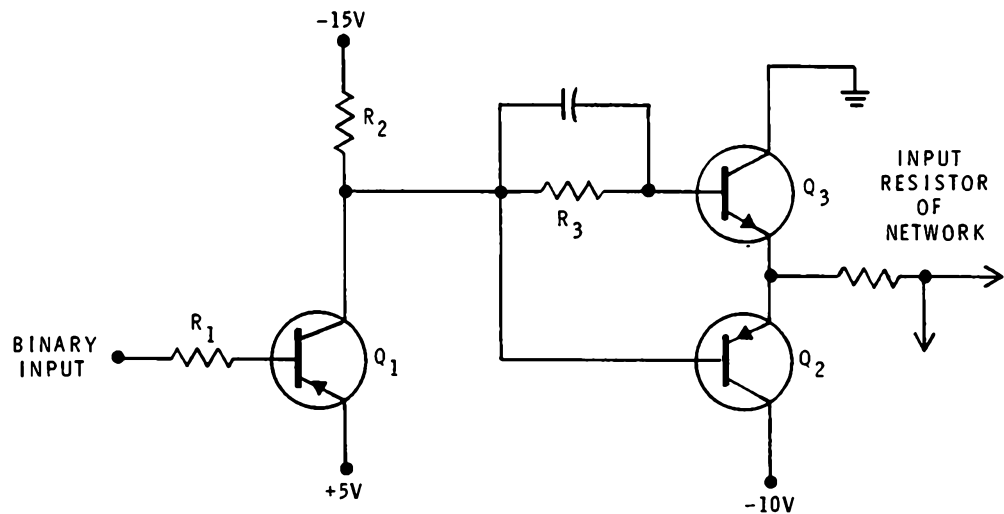


Figure 10-7

A bipolar inverted, overdriven complementary binary switch.

Figure 10-7 shows a typical electronic switch used for this application. This circuit is generally referred to as an inverted overdriven complementary emitter follower. Transistors Q2 and Q3 operate as switches to connect either the -10 volt reference or ground to the input resistor of the network. For example, when transistor Q2 conducts, -10 volts will be connected to the input. During this time, Q3 is cut off. When Q3 conducts and Q2 is cut off, the input resistor will be connected to ground through transistor Q3. Transistor Q1 is used as a driver circuit for the two switches and receives its input from one of the binary input bits. One of these complete switching circuits is required for each input bit.

Assume that the binary input is 0 volts. This causes transistor Q1 to become forward biased. With Q1 conducting, the output voltage at its collector will be approximately 5 volts. This $+5$ volts will be applied directly to the base of Q2 and to the base of Q3 through resistor R3. With this connection, Q2 will cut off and Q3 will conduct. Q3's very low conducting resistance will cause the input resistor to see 0 volts, or ground.

When the binary input is a binary 1 level of approximately 5 volts, Q1 will cut off. The -15 volt supply voltage will be applied to the base of Q3 through R3 and Q3 will cut off. Q2 will conduct and its low resistance will cause the -10 volt reference to be connected to the input resistor.

Transistors Q2 and Q3 are emitter followers configured as a complementary pair. The advantage of using emitter followers is that the saturation collector-to-emitter resistance is extremely low. Therefore, the collector-to-emitter voltage drop at saturation is also extremely low, only a few millivolts as compared to possibly hundreds of millivolts for a common-emitter configured transistor. This is important because the output voltage to the network resistor should be as close to ground or to -10 volts as possible. While the voltage drop across these two transistors can not be completely eliminated, in most cases the resultant error factor can be reduced to a point where it has little or no effect on circuit accuracy.

Improved Weighted Resistor DAC

The basic weighted resistor DAC can be further expanded to accept larger binary words. By simply adding an additional weighted resistor for each additional input bit, theoretically, any binary word size can be accommodated. However, there is a practical limitation to this circuit. The larger the number of input bits, the wider the range of resistor values required to produce the desired binary weighting. In the four-bit DAC discussed previously, the resistor values had an 8 to 1 range. Assume that we extended this to an 8 bit input word. The lowest value of resistor would still be R , but the highest value resistor would be $128R$. This represents a 128 to 1 ratio.

While a wide resistance range does not appear to be a problem, consider what happens when 12 bits are used. Suddenly, the ratio is 2048 to 1. With 10 k being the lowest value, the highest value would be 2,048 megohms. This is an enormous range of resistance values when it is necessary to have extreme accuracy. It is difficult to control the tolerance of resistors over such a wide range.

The accuracy of a D-to-A converter is highly dependent upon how precise its resistor values are. The wider the range of values, the more difficult the matching becomes. Other characteristics such as temperature tracking are also extremely difficult to control. Resistor values change with temperature, and different values of resistors change by different amounts. Therefore, over relatively wide temperature ranges, accuracy will decrease.

The temperature tracking problem can be overcome to some extent if a thin film hybrid fabrication of the resistance network is used. Alternatively, monolithic construction of the resistors in the form of an integrated circuit will greatly improve temperature tracking. However, when such integrated techniques are used, it is simply impossible to obtain very wide ranges of resistor values. The upper limit is approximately a 20 to 1 ratio in most integrated circuits, and since most D-to-A converters are in integrated circuit form today, obviously wide ranges of resistors values and weighted resistor arrangements are seldom used.

Another limitation of the basic weighted resistor circuit discussed earlier is the complexity and inaccuracy of the binary switch. These switches select either ground or a reference voltage, as indicated earlier. Because the transistor offsets cause errors, such circuits cannot be used in critical applications. Further, the basic switch described is not exceptionally fast. To overcome this limitation, special high speed current switches have been developed. These current switches, along with improved and modified weighted resistor networks, permit larger, faster, and more accurate DACs to be constructed.

Figure 10-8 shows a diagram of the modified weighted resistor DAC that overcomes the limitations mentioned above. This 12-bit DAC features three groups of binary weighted resistors. Each 4-bit group has identical values, 10 k, 20 k, 40 k, and 80 k ohms. For this reason, only an 8 to 1 resistance ratio exists. This is manageable in both hybrid and monolithic fabrication. These resistors are used in conjunction with bipolar transistors, connected as current source switches designated Q1 through Q12, in Figure 10-8. These current source switches are, in turn, driven by other current sources which form the buffering circuitry between the TTL digital input levels and the current switches.

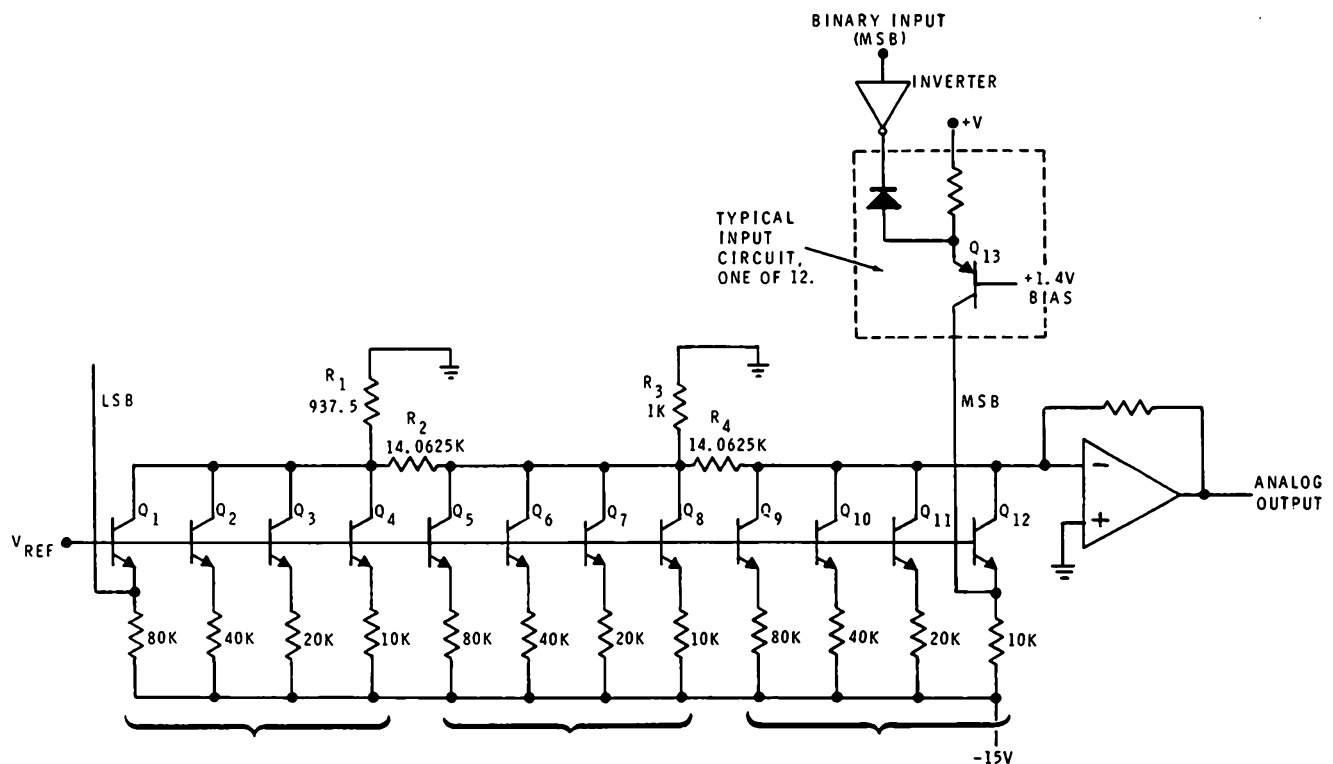


Figure 10-8

A 12-bit DAC using a modified weighted resistor network and current switches.

Now let's take a look to see how the circuit actually functions. First, if you look back to Figure 10-6, you can see that the simple 4-bit weighted resistor DAC consists of an accurate reference voltage connected to the weighted binary resistors. The resistors are switched in and out by the electronic switches. The voltage reference source, in combination with each resistor, forms four current sources. Since the summing junction on the op amp is effectively ground, each resistor can be assumed to generate a constant current, equal to the reference voltage, divided by the resistor value. A current source is simply an electronic source with a fixed value of current.

There are numerous ways in which electronic current sources are constructed. One of these is shown in Figure 10-9. A fixed voltage of +5.7 volts is applied to the base of the NPN transistor. The voltage is such that it will cause the emitter base junction to be forward biased.

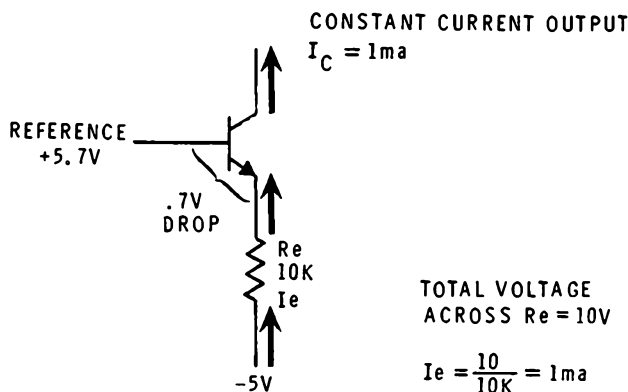


Figure 10-9

A constant current source.

The voltage appearing across the emitter resistor is the base bias voltage +5.7 less the emitter base voltage drop of .7 volts plus the -5 volt emitter supply or +10 volts. The voltage across the emitter resistor divided by the value of the resistor gives us the emitter current; for this circuit, 1 mA. Recall that in most high gain transistors, the collector current is very nearly equal to the emitter current. The collector output, then, is a fixed 1 mA current source. By varying the size of the resistor, other values of current can be obtained.

Refer back to Figure 10-8. Using this concept, four binary weighted current sources are constructed using NPN transistors, the binary weighted resistors, and an accurate voltage reference source. The output of these current sources are connected together and applied to the summing junction of the op amp. The current source transistors can be turned off and on by the binary input signals. Therefore, the current contributed to the summing junction will be proportional to the binary input value. We will show in a minute how the current sources are turned off and on.

In Figure 10-8, three groups of the 4-bit weighted resistor networks are used to accommodate a 12-bit binary input word. With the same resistor values and reference voltages used in each of the three sets, each will contribute the same amount of current. In order to cause the least significant bits to have less binary weight than the more significant bits, additional resistors are added to the network. Note resistors R1 through R4. R1 and R2 form a voltage divider circuit that effectively provides 16 to 1 current attenuation for the second group of four sources. R3 and R4 provide another 16 to 1 current attenuation. With this arrangement, the proper weighting occurs. Additional groups of four can be added to expand the binary input word to 16 or 20 bits.

If the resistor values R1 and R2, and R3 and R4, are selected for a 10 to 1 current reduction, the circuit in Figure 10-8 can be used to accommodate BCD inputs. Then, each 4-bit BCD input would control one of the 4-bit weighted resistor groups. While BCD DACs are not as widely used as DACs, they do find applications in some special equipment.

Now refer to Figure 10-8 again. Let's see how the binary input word is buffered by the current sources to switch the desired current source switches in and out. Consider the most significant input bit. This bit is applied to an inverter, which in turn drives a diode connected to the emitter of the current source. TTL input levels are assumed. PNP current source transistor Q13 is biased to the midrange of the typical TTL voltage level of approximately 1.4 volts. First, assume a binary 1 input. The output of the inverter will then be binary 0. This will cause the cathode of the diode to be essentially at ground. The emitter of Q13 is approximately 2 volts; therefore, the diode is forward biased. The conducting diode effectively shunts emitter current away from Q13. Therefore, Q13 is cut off and has no effect on Q12. Q12 then conducts and causes current to be injected into the summing junction of the op amp.

If the input is a binary 0, the output of the inverter would be about +3 volts. The diode would then be reverse biased. This causes current source Q13 to divert current flow away from current source Q12. Therefore, Q12 is cut off and supplies no current to the op amp summing junction.

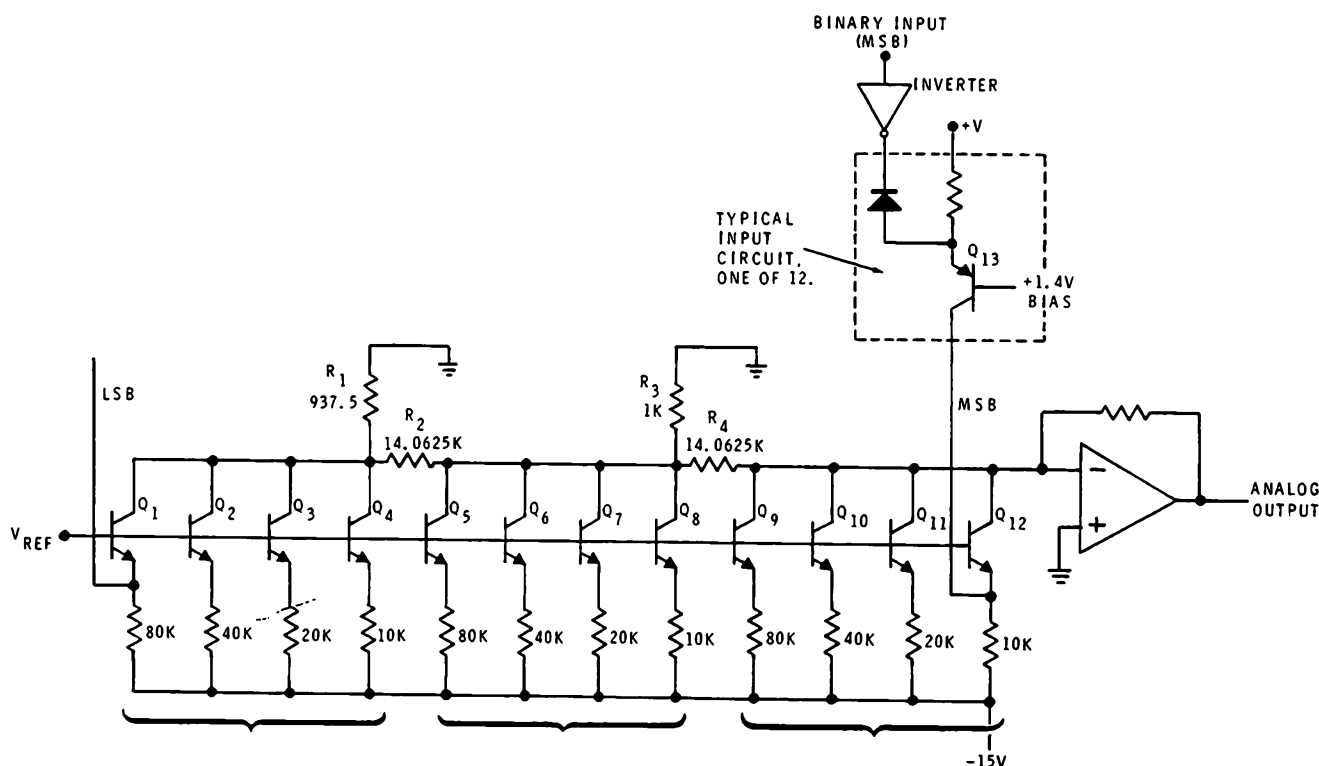


Figure 10-8

A 12-bit DAC using a modified weighted resistor network and current switches.

R/2R Ladder DAC

While the weighted resistor DAC is widely used, another resistor network is more commonly found in DACs. Known as the R/2R ladder, this resistor network overcomes the wide resistance range limitation normally associated with the weighted resistor network. Only two values of resistance are used, R and 2R.

Figure 10-10A shows the basic R/2R ladder configuration. Like the other networks, the R/2R ladder generates an output current. This is usually converted into an output voltage with an op amp as shown. Binary controlled electronic switches connect each 2R resistor to either a reference voltage or ground. The result is a binary weighted current at the summing junction of the op amp. The output is proportional to the binary input.

Another version of the R/2R network is shown in Figure 10-10B. Here the reference is applied to the network, and the electronic switches connect the 2R resistor to either ground or the summing junction. This form of the R/2R ladder is often referred to as the inverted R/2R. It is by far the most widely used of the two versions. The main advantage of this version is that the switches can be current operated, rather than voltage operated as in the first version.

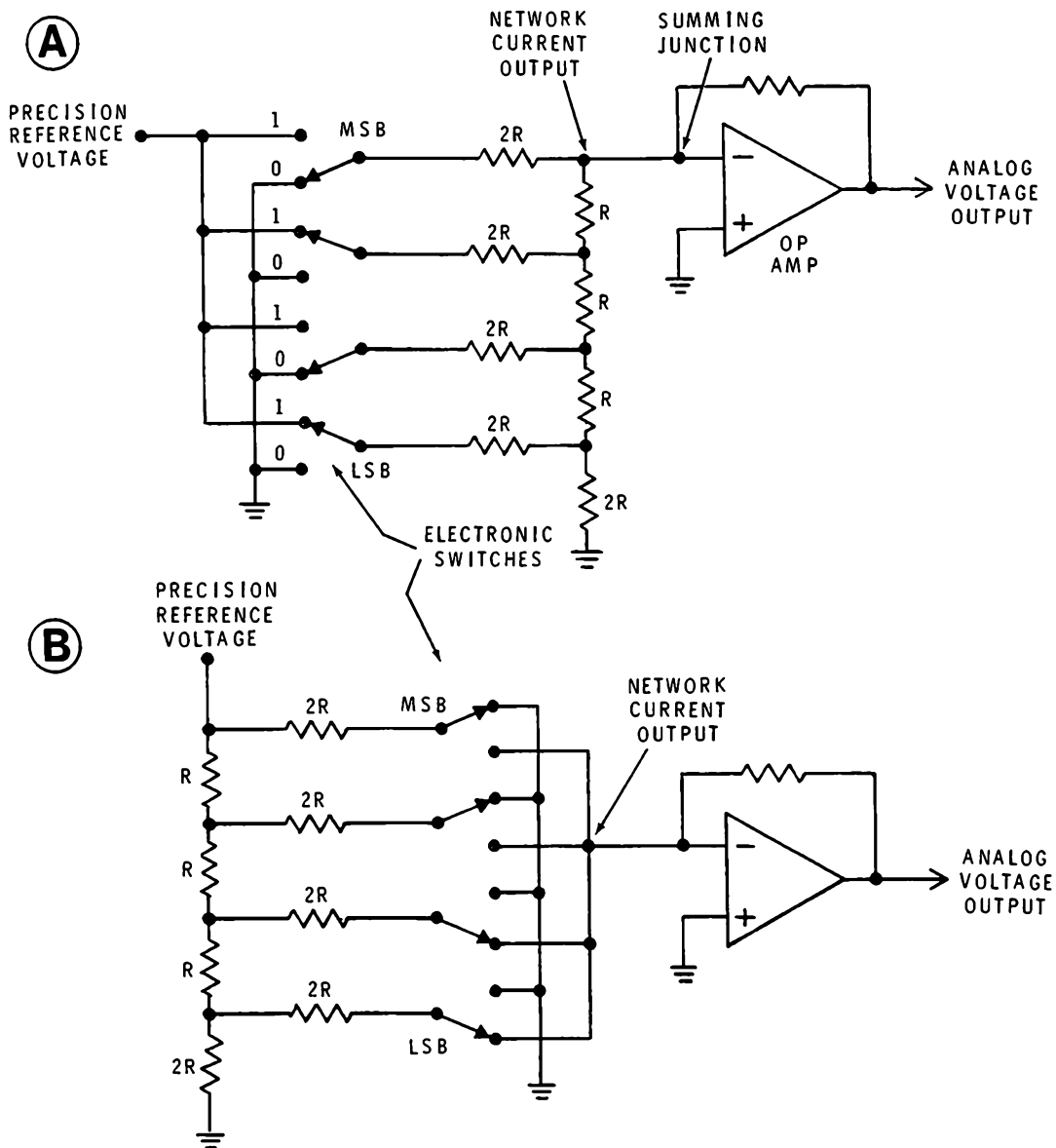


Figure 10-10

R/2R network DACs.
(A) standard (B) inverted

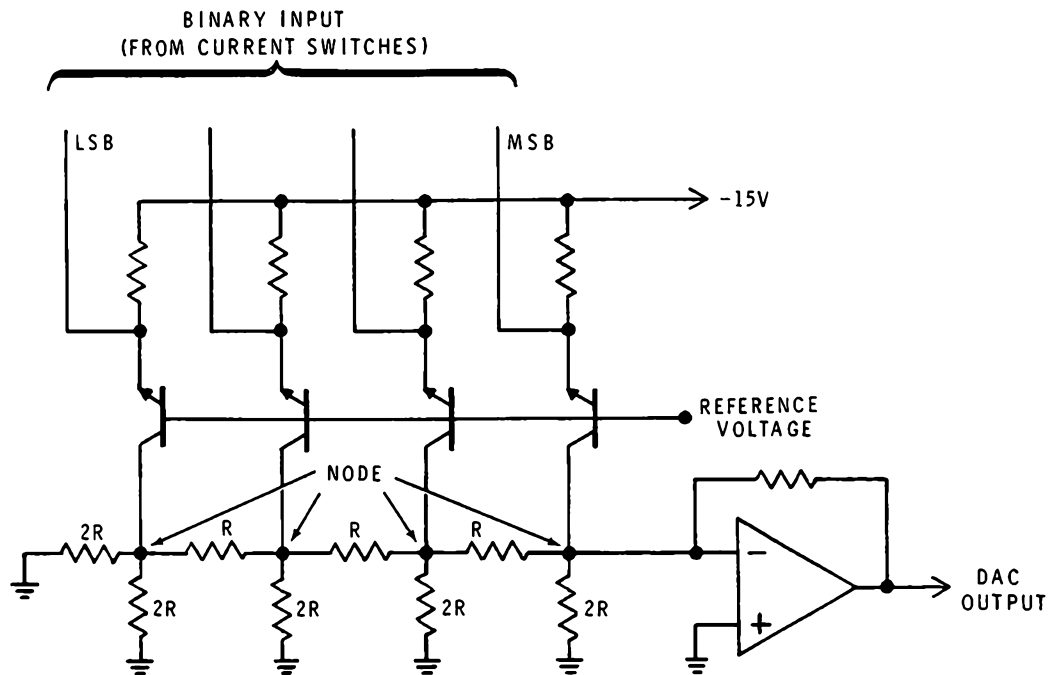


Figure 10-11

An $R/2R$ DAC using current switched current sources.

An alternative method of using the $R/2R$ network is shown in Figure 10-11. Here the $2R$ values of the network are grounded. Equal value current sources are then connected to each node of the $R/2R$ network. These current sources are then switched in and out of the ladder by the binary input signals. This arrangement of switching the current into the node of the network produces exactly the same effect as switching the $2R$ resistance values to either ground or a reference voltage, or from ground to the summing junction.

Current switches similar to those shown in Figure 10-8 are used to buffer the TTL input from the current sources.


The reason for using this switching technique is that it is usually faster to switch a current than a voltage. Higher data conversion rates can thus be achieved. Further, current switching usually lends itself to integrated circuit construction better than do most voltage operated switches.



Typical DACs

Virtually all DACs in use today are integrated circuits. A wide range of IC DACs are available. DACs with 6- to 20-bit input words can be obtained in a wide range of accuracies and conversion times. The most popular DAC sizes are 8, 10, 12, and 16 bits. Advanced integrated circuit manufacturing techniques have allowed all DAC circuitry to be made on a single silicon chip. Such monolithic DACs contain the resistor network, logic switches, current switches, and reference source; and in some cases the operational amplifier used to convert the current output into a voltage output.


While full monolithic DACs serve a wide range of purposes, even higher quality DACs can be had with hybrid integrated circuit techniques. Hybrid techniques combine both monolithic circuitry with other types of integrated circuit techniques such as thin film. In hybrid DACs, thin film resistor networks are used instead of monolithic networks. Manufacturing a high precision resistor network with monolithic techniques is extremely difficult. However, with thin film techniques, high precision resistor values are more easily obtained.



Thin film resistor networks are made of nickel-chrome, silicon-chrome, or some other similar alloy. This alloy is deposited on the IC substrate. The precisely controlled deposition techniques allow very precise resistor values to be obtained. These are further trimmed to exact value with a laser. The fine, high-intensity laser beam burns away selected portions of the thin film deposits to set the resistor values to their optimum value. At the present time, such precision cannot be obtained using monolithic techniques.

While hybrid DACs are more accurate, they are also more expensive, because of the complexity of combining thin film and monolithic circuitry in the same package. Nevertheless, such precision is required for many applications.

Now let's take a look at a typical DAC. We have selected the Motorola MC-1406/7/8 series. This is a fully monolithic DAC available in 6-, 7-, and 8-bit resolutions.



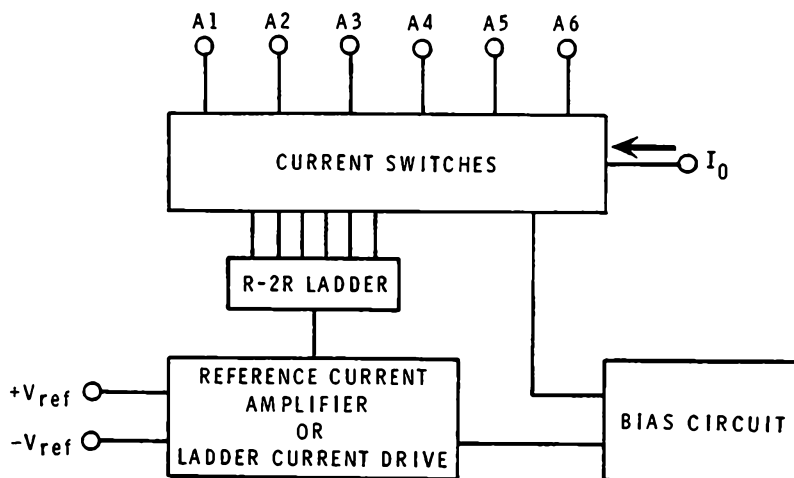


Figure 10-12

Block diagram of MC1406L.

Figure 10-12 shows a general block diagram of the 1406 DAC. An R/2R resistive ladder is used for the basic conversion. The reference current amplifier circuit furnishes current to the resistive network. Current switches, under the control of the binary input word, switch the current into and out of the various R/2R ladder legs. The resulting output is a binary weighted current I_0 . The bias circuit generates the necessary bias voltages and currents for the current switches. An external op amp is normally used to convert the current output into a voltage output.

Figure 10-13 shows the detailed schematic diagram of the 1406 DAC. Note that the R/2R ladder is made up of 400 and 800 ohm monolithic resistors. The reference current amplifier circuitry injects a fixed current into the left-hand node of the R/2R ladder. The value of this current is set by an external reference voltage and resistance. This can be any value in the 0 to 2 mA range.

The R/2R ladders network divides and scales this current into six binary weighted levels. The binary weighted current level from each 800 ohm resistor in the ladder network is switched to a common I_0 current bus. Switching is accomplished by a bipolar switching circuit that is controlled by the binary input. The binary weighted output current is fed to the summing junction of an op amp, which produces an output voltage proportional to the binary input. The op amp feedback resistor scales the output voltage to the desired level.

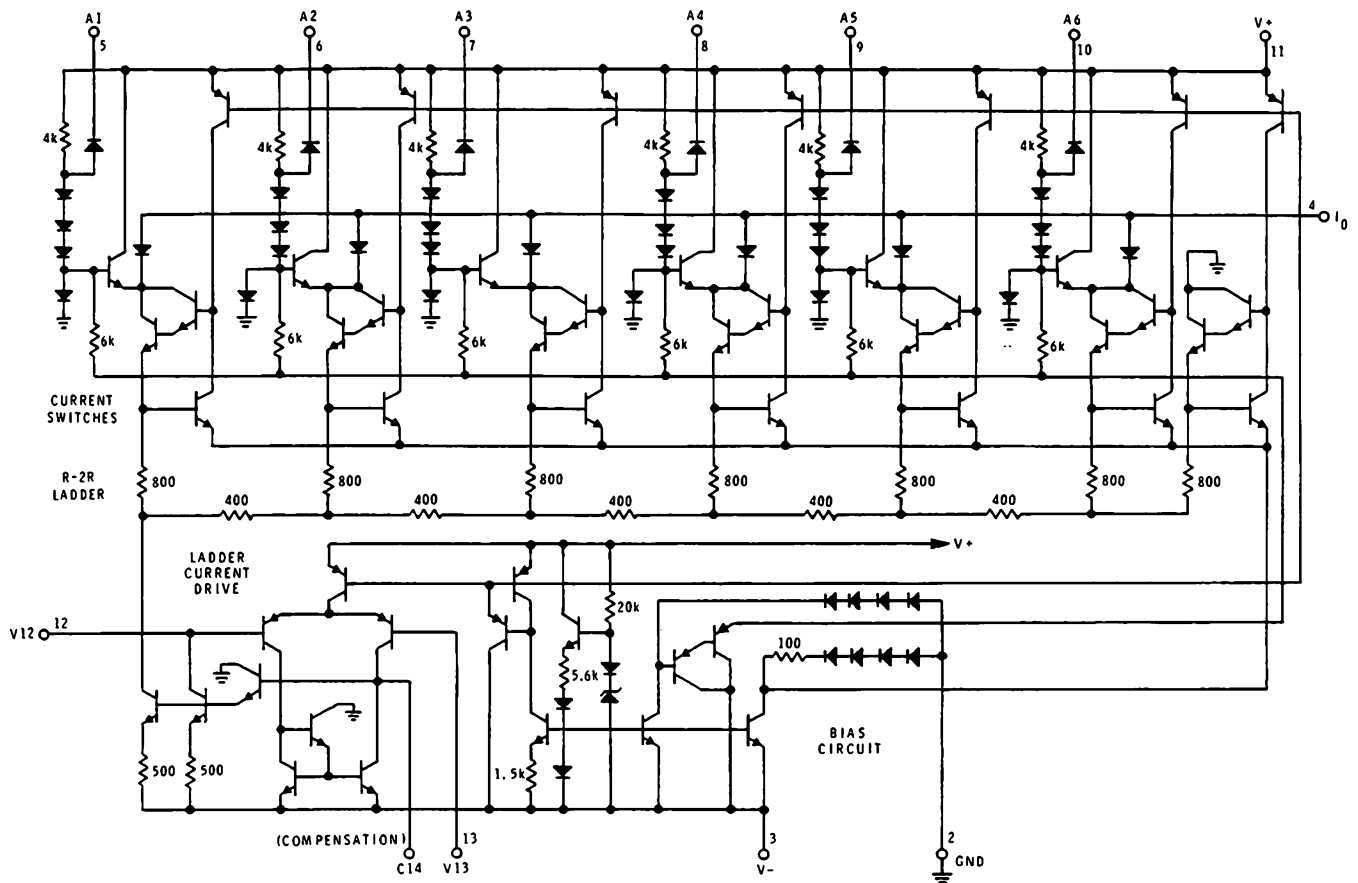
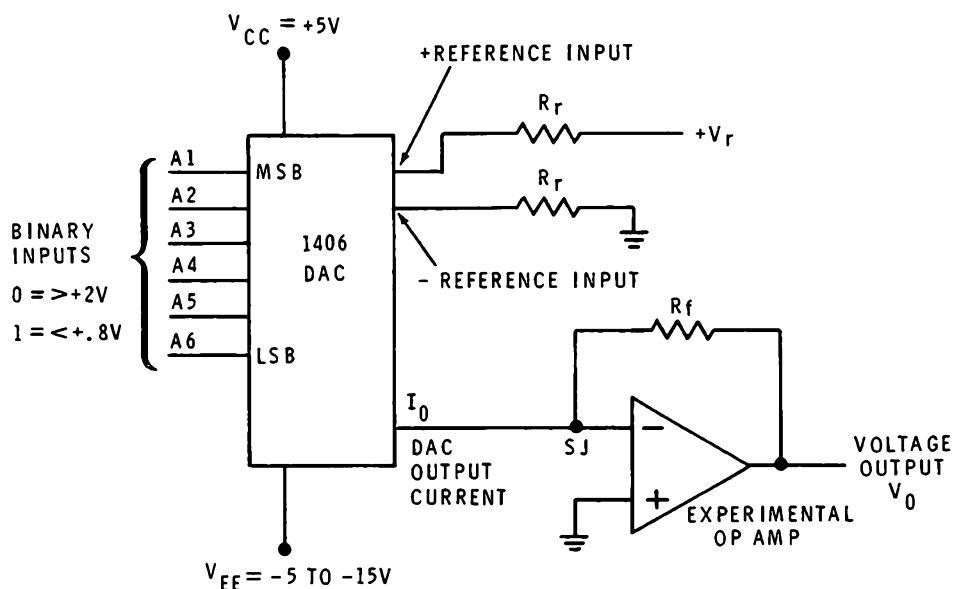


Figure 10-13

Complete circuit schematic.

Figure 10-14 shows a simplified block diagram of the 1406 DAC. This is the way you will see the DAC portrayed in schematic diagrams. There are a couple of important things to notice. First, the DAC is powered by two supplies, V_{CC} , a +5 volt supply, and V_{EE} , a negative supply that is usually in the -5 to -15 volt range.



$$V_0 = \frac{V_r}{R_r} (R_f) \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} \right)$$

WHERE A1-A6 IS 1 WHEN THE BINARY INPUT IS BINARY 1 (<+.8V).

Figure 10-14

Schematic diagram of a 1406 DAC
showing input and output connections.

There are six binary inputs, A1 - A6, that are TTL compatible. However, they use negative logic, where +2 volts or more input is a binary 0 and +.8 volt input or less is a binary 1.

There are two reference inputs, one positive and one negative. The DAC current output will be positive with a negative reference input or negative with a positive reference input. In Figure 10-13, the positive reference is used. The reference current, I_r , is set by the value of the reference voltage, V_r , and the external resistor, R_r .

$$I_r = V_r/R_r$$

The reference current is in the 0 to 2 mA range. Setting the reference current sets the maximum I_o current, which is 63/64 of the reference current. The unused negative reference input is connected to ground through a resistor value equal to R_r .

The DAC output I_o is connected to the summing junction (SJ) of an external op amp. Almost any common IC op amp can be used. Its feedback resistor R_f will scale the output voltage to the desired value. The output voltage will be $V_o = I_o \times R_f$. Remember, the op amp inverts. Therefore, for a negative input current, the output voltage will be positive as shown in Figure 10-14.

You can compute the actual value of the output voltage using the formula shown at the bottom of Figure 10-14.

Finally, the maximum conversion error is plus or minus .78%. The conversion speed, a function of circuit propagation delay and settling time, is in the 150 to 300 nanosecond range.

DAC Specifications and Error Sources

There are certain characteristics that determine the quality of a digital-to-analog converter. The specifications that influence the quality of the circuit are: resolution, accuracy, linearity, monotonicity, and settling time. Let's consider each of these specifications in more detail.

RESOLUTION.

Resolution refers to the number of discrete output voltage levels the DAC is capable of. This, in turn, is a function of the number of input bits. As an example, consider an 8-bit DAC. With eight bits, the DAC can resolve 2 to the 8th power, or 256 discrete output voltage levels. We usually say that an 8-bit DAC has a resolution of 1 part in 256.

A 12-bit DAC has 2 to the 12th power, or 4096 discrete output voltage levels. The resolution of a 12-bit DAC, then, is 1 in 4096. Sometimes the resolution is expressed as a percentage. For example, the resolution of an 8-bit DAC would be $1/256 = .0039$ or .39%. The resolution of a 12-bit DAC would be $1/4096 = .00024$ or .024%.

In terms of voltage steps, assume an input reference voltage of 10 volts. An 8-bit DAC then would have 256 equal output voltage segments of approximately 39 millivolts each. The 12-bit DAC with a 10 volt reference would produce 4096 discrete output voltage levels between 0 and 10 volts in 2.4 millivolt increments.

You will also see resolution referred to in terms of parts per million (PPM). This method of measurement assumes that the output (zero to the reference voltage) is divided into one million parts. The resolution is expressed in number of millionth parts that make up the minimum incremental change produced by the LSB. The PPM value is obtained by multiplying the fractional expression of percentage by 1,000,000. For example, the 8-bit DAC has a resolution of $1/256$, or .0039, as indicated earlier. Multiplying this by 1,000,000 gives a resolution of 3900 PPM.

ACCURACY.

Accuracy, or relative accuracy, is a measure of the deviation from an ideal theoretical output, usually the full scale output current or voltage. This accuracy is usually expressed as a percentage and represents the error percentage rather than the accuracy percentage (actual value = accuracy + error). For example, the relative accuracy of the popular 1406 6-bit monolithic DAC is .78%. This is the error relative to the full-scale current output.

It is important to note that different manufacturers express accuracy in different ways. It is desirable to refer to the manufacturer's data sheet and application notes. Be sure that you fully understand their definition of accuracy.

Consider the relative accuracy of a DAC with a rating of .3%. Given a 10-volt full scale output, the accuracy then would be 10×0.003 or 30 millivolts. What that really means is that the actual output will be within 30 millivolts of the ideal or true output.

Note also that the accuracy figure should be less than the resolution expressed as a percentage. If the accuracy, or rather the error, figure is larger than the resolution figure, then the least significant bit changes are essentially meaningless.

LINEARITY.

Linearity, or linearity error, is the amount that the actual output of a DAC differs from an ideal straight line output. While the output from a DAC is in the form of discrete steps or increments of output voltage; theoretically, the output should be a perfect straight line from 0 volts to full scale output voltage if an infinite number of steps were used. Linearity is the error that exists between this theoretical straight line curve and the actual discrete output steps. The maximum allowable deviation from the straight line is less than one-half the magnitude of one theoretically perfect step. This is usually referred to as + or - one-half the LSB, or the error is + or - one-half the voltage resolution of the least significant bit. See Figure 10-15.

The linearity of a DAC is primarily a function of the precision of the resistors in the network and the various voltage drops associated with the current switches. Unequal values of resistors or different switch offset voltages cause minor linearity errors.

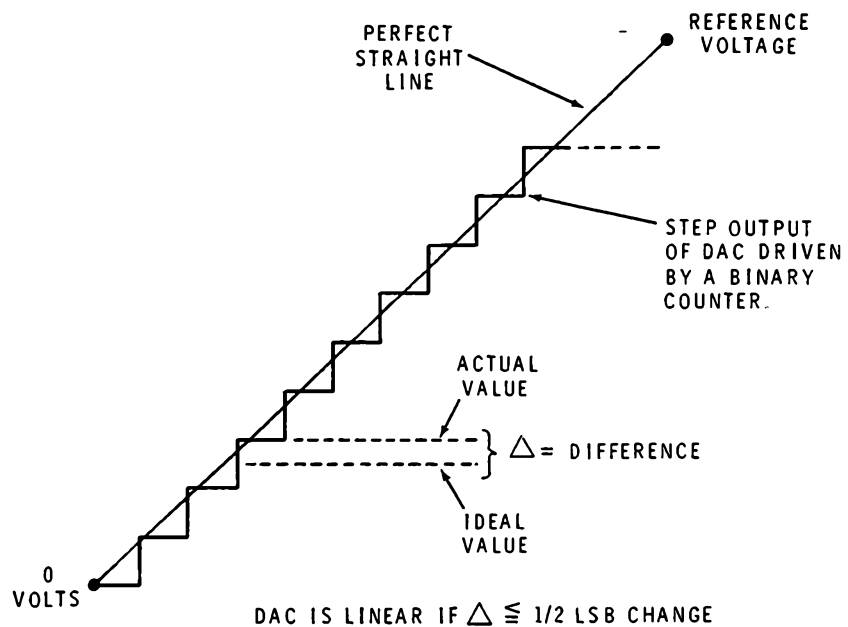


Figure 10-15

DAC linearity.

MONOTONICITY.

A DAC is said to be monotonic if its output increases or remains the same for an increasing binary input count. For example, if a DAC is connected to a binary counter which is incremented, the DAC output should increase by one step of resolution for each input count. If the output of the DAC misses a step or steps backward (drops in output) for an increment of binary count, the DAC is said to be nonmonotonic. These two cases are illustrated in Figure 10-16.

Monotonicity, as you can see, is related to linearity. The lack of monotonicity can be caused by poor matching in the resistive network and excessive variations in the current switching networks.

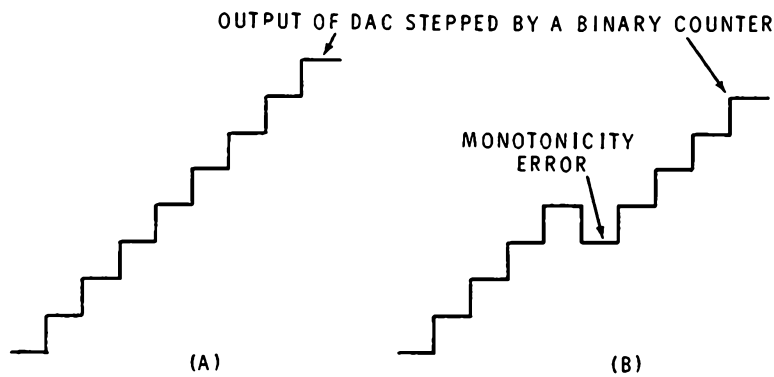


Figure 10-16

Monotonicity.

(A) monotonic output;

(B) non-monotonic output.

SETTLING TIME.

Settling time is the specification that refers to the amount of time required by the converter output to settle within a given range of output voltage after a binary input word change. As you might suspect, if a binary input change occurs, the current switches will switch selected network resistors in and out of the circuit. The resulting voltage and current changes will take time to settle down. The inductances and capacitances in the circuit will cause a certain amount of ringing or overshoot.

Settling time is normally defined to be the period of time required for the DAC output to settle within $\pm 1/2$ LSB of the final value. Recall that the LSB value refers to the minimum increment of resolution. Figure 10-17 shows settling time with reference to an output change.

The settling time is, in effect, a measure of how fast a DAC can operate. You might say it is a measure of its conversion time. The settling time indicates how fast binary input changes can occur and still produce output voltages within the accuracy rating of the DAC. The propagation delay of the digital circuit also adversely affect conversion time. Typical modern DACs are extremely fast and have settling times in the 20 to 500 nanoseconds range.

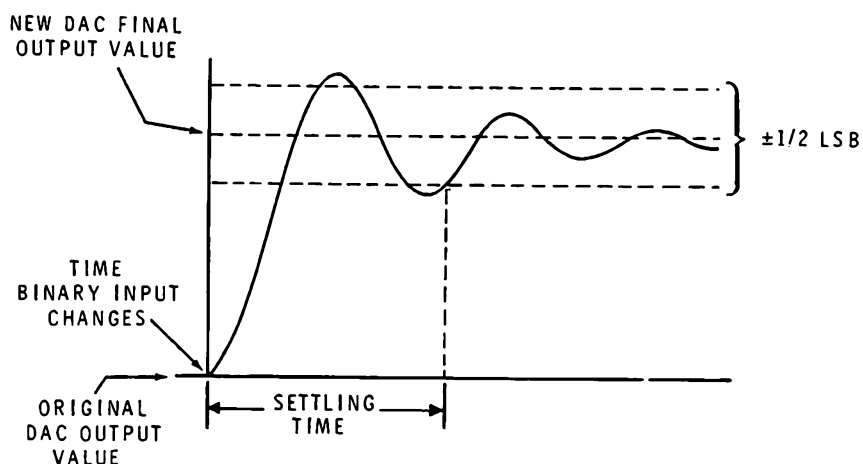


Figure 10-17

Settling time.

Multiplying DACs

In our discussion of DACs we have always assumed that the magnitude of the reference voltage was fixed. Most DACs use a constant reference voltage, and the output range of the DAC is determined by the value of that reference voltage. However, there is no reason why a DAC can't be constructed with a variable input reference voltage. The output of the DAC is directly proportional to the binary input value and the magnitude of the reference voltage. In fact, the output is the product of the reference voltage and the binary input. Because of this product relationship, DACs with variable references are usually referred to as multiplying DACs. You will also hear them referred to as hybrid multipliers, where hybrid in this case refers to the mix of analog and digital techniques.

You will also hear multiplying DACs referred to as digital attenuators. The multiplying DAC acts like a digitally controlled potentiometer. An analog signal is applied to the DAC reference input. The binary input specifies an attenuation factor. With a 0 binary input, the output of the multiplying DAC or digital attenuator would be zero. With a full scale binary input (all binary 1's), the output of the DAC would be the full reference voltage. Binary input values between zero and the maximum input value will produce intermediate levels of attenuation for the analog input signal. The binary input specifies an attenuation value between 0 and 1.

The ability of the electronic switches associated with the resistive network determines whether or not the DAC can be used in a multiplying application. Most DAC switches are designed for a fixed polarity of reference voltage. While they may be used as a multiplying DAC, the analog or reference input voltage must be of the desired polarity. Such multiplying DACs are referred to as uni-polar devices. Uni-polar multiplying DACs usually have bipolar transistor network switches.

Bipolar analog input signals can still be accommodated by some multiplying DACs. This means AC signals with both input polarities can be accommodated. Such DACs normally use field effect transistor switches, which can deal with either polarity signal.

Figure 10-18 shows the symbol used to represent a multiplying DAC (MDAC) and a bipolar application. Here the MDAC is stepped by a binary counter. The output is an attenuated but stepped amplitude version of the input.

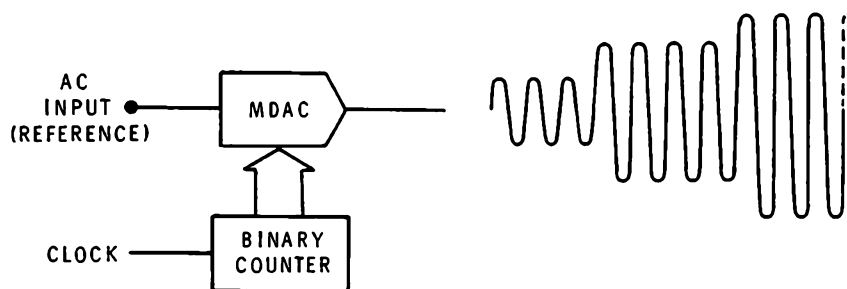


Figure 10-18

A multiplexing bipolar DAC.

Self Test Review

1. The three main parts of a DAC are:
 - a. _____
 - b. _____
 - c. _____
2. The part (above) that actually performs the conversion is the _____.
3. The main disadvantage of the weighted resistor DAC network is its:
 - a. slow settling time
 - b. narrow resistor ranges
 - c. size
 - d. wide resistance ranges
4. The output of most DACs is a:
 - a. current
 - b. voltage
5. The op amp in a DAC is used for _____ to _____ conversion.
6. The DAC network with the narrowest resistor range is called the _____ ladder.
7. The three types of DAC switches are:
 - a. _____
 - b. _____
 - c. _____

8. BCD inputs can be accommodated by using a/an _____ DAC.
9. The two types of IC DACs are _____ and _____.
10. Typical IC DACs are available in bit sizes of _____, _____, _____, _____, and _____ bits.
11. Express the resolution of a 10-bit DAC with a 10-volt reference in each of the following terms:
- _____ %
_____ volts
_____ PPM
12. If the output of a DAC that is stepped by a counter increases with each input step, the DAC is said to be _____.
13. The error between a DAC output and a theoretical straight line is a measure of the _____ and is expressed in terms of _____.
14. Conversion time is primarily a function of the _____ and the _____ of a DAC.
15. Another name for a digital attenuator is _____.

Answers

1.
 - a. Resistor network
 - b. Reference
 - c. Electronic switches
2. Resistor network
3. d. wide resistance range
4. a. current
5. current to voltage
6. R/2R
7.
 - a. mechanical
 - b. voltage (either current or voltage)
 - c. current
8. improved weighted resistor network
9. monolithic, hybrid
10. 6, 8, 10, 12, 16, 20
11. .09765%
.9765 millivolts/step
976.5 PPM
10 bits = 1024 parts
12. monotonic
13. linearity, 1/2 LSB voltage
14. settling time, propagation delay
15. multiplying DAC

EXPERIMENT 24

Digital-To-Analog Conversion

OBJECTIVES: *To demonstrate the operation and application of a modern LSI digital-to-analog converter.*

Introduction

In this experiment you are going to demonstrate the operation of an IC DAC. You will actually build a complete digital-to-analog converter using the Motorola 1408 DAC. Although you read about the 1406 DAC in Unit 10, the only significant difference between them is the number of inputs — the 1406 has 6, and the 1408 has 8. Parallel binary inputs from various sources will be applied to the DAC which, in turn, converts the binary numbers into a proportional output current. An op amp is used to convert the DAC current output into an appropriate voltage. The op amp feedback resistor is varied to scale the output to the desired level.

Material Needed

Heathkit Digital Design Experimenter
Voltmeter capable of measuring 0 to 15 volts DC
Oscilloscope
1—MC1408 DAC IC (442-751)
1—1458 Dual Op Amp IC (442-21)
1—14495-1 LED Driver IC (443-1802)
2—74LS193 TTL IC (443-815)
1—7-Segment LED (411-885)
1—5 k Potentiometer (Control)
1—10 k Potentiometer (Control)
4—47 k Resistor
1—.1 μ F Ceramic Capacitor

Procedure

1. With the ET-3200 Trainer off, assemble the circuit shown in Figure 10-19. Install all of the circuit components as close to the right end of the breadboard as possible. You will be installing more components later in the experiment. Double-check all of your connections before you apply power.

In this first portion of the experiment, you will be manually switching the binary inputs to the DAC. Due to the limits of the Trainer, we cannot provide you with eight separate binary switches. Notice that in Figure 10-19, inputs D4 through D7 are tied low, and inputs D0 through D3 are tied to logic switches SW1 through SW4. You'll begin by setting the four logic switches high, and adjusting the feedback resistor of the op amp to an established voltage. Once this is accomplished, the op amp will step from 0 volts to the established voltage in 15 steps—the highest count for the four logic switches.

SW4 is the LSB for our binary input, and SW1 is our MSB. The LSB for any counter connects to input D0 on the DAC, and the MSB—when there are 8 binary inputs—connects to D7. In this configuration, the MSB is located at input D3.

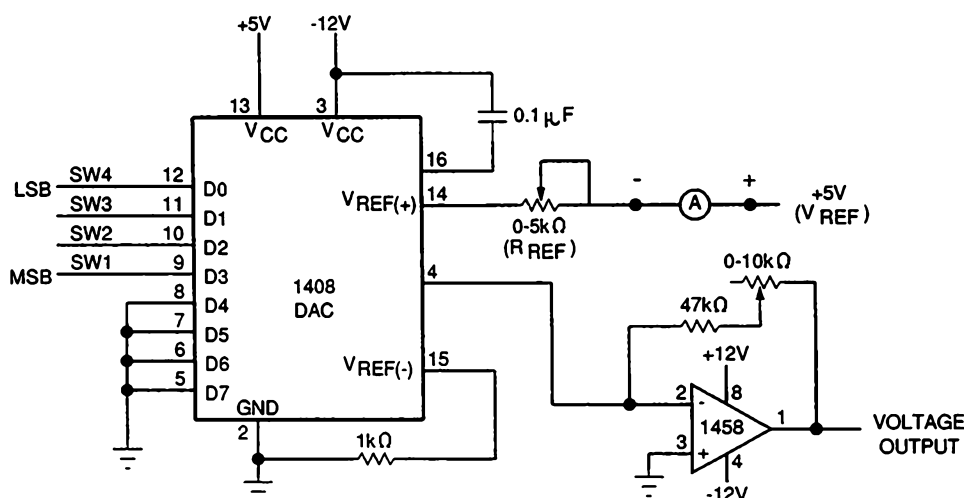


Figure 10-19

2. Place an ammeter in the position shown in Figure 10-19, and set all the data switches to the down position.
3. Apply power to the circuit, and adjust the 5 kilohm control until the current through the ammeter reads 2 milliamps (2 mA). This sets the reference voltage to its proper level.
4. Once you have adjusted the current level, turn the power off and remove the ammeter from the circuit, replacing it with a jumper wire.
5. Reapply power to the circuit and connect a DC voltmeter to the voltage output of the circuit (pin 1 of the 1458 op amp). Move all the data switches to the up position, and adjust the 10 kilohm control on the feedback path of the op amp until the output reaches +6.0 volts.
6. Move all the data switches to the down position, and verify that the output voltage is 0 volts. Answer the following questions:

- a. With the data switches in their down position, the decimal input is 0, and the output voltage is 0 volts. When all the data switches are in their up position, what is the binary and decimal input?

Binary input = _____

Decimal input = _____

- b. How many steps are required for the DAC to go from 0 volts to +6 volts?

_____ steps

- c. Based upon your answer to b, the output voltage will increase by what amount for each step?

_____ V

7. Using Table I, step the data switches through all the binary inputs shown. For each binary input, write the equivalent decimal input and the output voltage value. Remember that SW4 is the LSB input, and SW1 is the MSB.

Table I

Binary Input	Decimal Input	Output Voltage
0000		
0001		
0010		
0011		
0100		
0101		
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111		

8. Turn the Trainer off. Leave the circuit connected, as you will be using it in the next portion of the experiment.

Discussion

The circuit you constructed in this portion of the experiment is a complete DAC using the 1408 IC. The 1408 uses both +5 and -12 volt supply voltages, and a reference input current. In Step 3, you adjusted the 5 kilohm control to set the reference input current to 2 milliamps. This reference current determines the output current value.

The output current is fed to the inverting input of the 1458 op amp, which acts as a current-to-voltage converter. The feedback resistance is determined primarily by the 47 kilohm resistor, but we added the 10 kilohm control to allow you to adjust the level of the output voltage. In Step 5, you set the binary input to 1111 (the highest available binary input), and set the output voltage to +6 volts. You then ensured the output voltage dropped to 0 volts when the binary input was 0000.

Step 6 required you to determine the stepping voltage of the circuit. When the binary input is 1111, the decimal input is 15. It would therefore require 15 steps for the binary input to go from 0000 to 1111. A binary input of 0000 (decimal 0) causes an output voltage of 0 volts, and 15 steps later, a binary input of 1111 (decimal 15) causes an output of +6 volts. Therefore, for each binary step, the output increases by:

$$\frac{6 \text{ V}}{15 \text{ steps}} = 0.4 \text{ V/step}$$

In Step 7, you verified that the output voltage stepped by 0.4 volts by measuring the output voltage for each binary input. As you completed Table I, you should have seen the output voltage increase from 0 volts to +6 volts in 0.4-volt increments.

When you manually switch the binary inputs, you can see the DAC work at its most primary level. However, we only used 4 of the possible 8 binary inputs, allowing the output to increase from 0 volts to our established voltage in only 15 steps. In the next portion of the experiment, all 8 binary inputs will be tied to two counters, which will automatically switch the input. If this switching occurs at a high enough frequency, you will be able to see the ramp-like output typical of the DAC.

Procedure (continued)

9. With the Trainer still off, wire the circuit shown in Figure 10-20A. The circuit you wired in the previous steps will remain the same, except the 47 k Ω feedback resistor is replaced by a 1 k Ω resistor. Also, you are substituting the 4 data switches with the two cascaded counters shown. As before, position all of the circuit components as close to the right end of the breadboard as possible. You will use this circuit, plus additional components, in a later experiment.

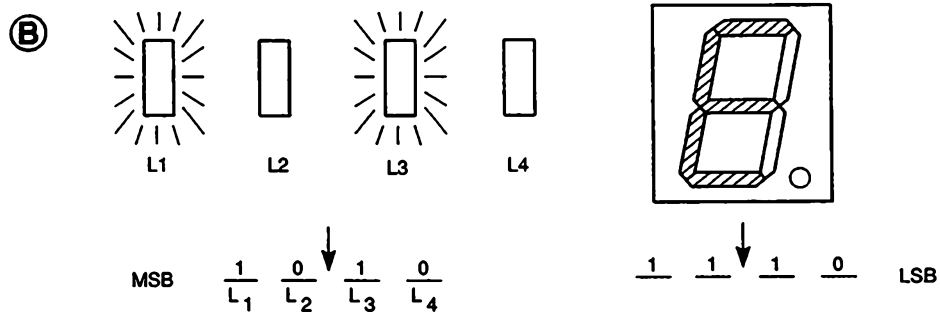
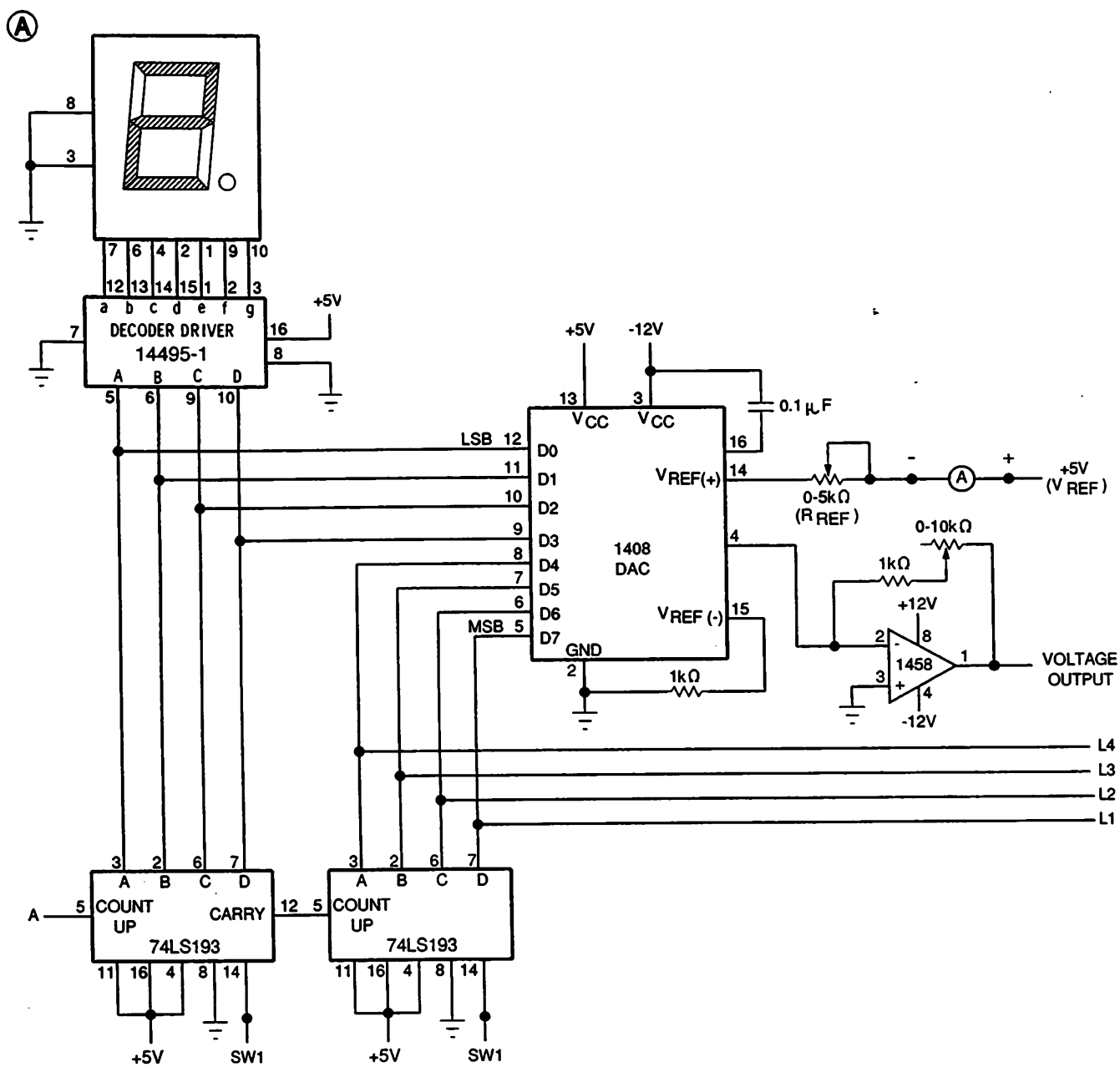


Figure 10-20

The two counters will provide the 8-bit binary input to the DAC circuit. Counter #1 provides the inputs for the four least significant bits of the DAC. Its binary count is also sent to a BCD-to-7-segment decoder/driver and a 7-segment LED. The carry signal from counter #1 (pin 12) is used as the clock input for counter #2. Counter #2 supplies the binary inputs for the four most significant bits of the DAC. By wiring the outputs of counter #2 to the four logic indicators on the Trainer, you can view the output for counter #2.

To determine the binary input to the DAC, simply convert the 7-segment hexadecimal readout to binary for the four LSB inputs, and the logic indicators on the Trainer (L4 through L1) for the four MSB inputs. In Figure 10-20B, the 7-segment display shows an "E," while L1 and L3 are lit. The "E" translates to a decimal 14, which is a binary 1110. Meanwhile, L4 through L1 read right-to-left for their binary count. The result is the binary input 10101110. (Which is, by the way, a decimal count of 174.) To make things simpler, you could remember that for every sixteen counts of counter #1 (displayed by the 7-segment LED), counter #2 counts once.

The overall count for these cascaded circuits is 2^8 , or 256. Actually, since one of these counts recycles the counter to zero, there are 255 incremented steps.

You will begin by connecting the clock input to counter #1 to logic switch A, as shown in Figure 10-20A. Later, you will be moving this input to the CLOCK connector on the Trainer. It is important that the counters be locked to their reset condition while you are moving the clock input. Notice that the reset input for both counters (pin 14) is tied to data switch #1. If the data switch is kept low, the counters will count normally. However, if SW1 is set high, the counters will lock to a low output. You will set SW1 high anytime you move the clock input, and then return the switch to its original position.

10. To begin, data SW1 should be high, and the CLOCK input should be connected to logic switch A.

11. Apply power to the circuit, and adjust the 5 kilohm control until the current through the ammeter reads 2 milliamps (2 mA). Once this is completed, turn the power off and remove the ammeter from the circuit, replacing it with a jumper wire.
12. Reapply power to the circuit and connect a DC voltmeter to the voltage output of the circuit (pin 1 of the 1458 op amp).
13. Move data SW1 to its low position. Toggle logic switch A until L3 is lit (all others off), and the 7-segment LED reads "8."
14. Adjust the 10 kilohm control on the feedback path of the op amp until the output voltage equals +1.60 volts. Answer the following questions:

a. What is the present binary input to the DAC?

b. What is the decimal input?

c. If you adjusted the output voltage to +1.60 volts for this binary and decimal input, by what amount would the output voltage change in a single step?

_____ V

15. Set data SW1 high (resetting the counters), then low again. Toggle logic switch A a random number of times, and note the amount of voltage change for each step. Record your measurement below:

Voltage change-per-step = _____ V

Is this voltage equal to your answer for Question "c" back in Step 14?

_____ (yes/no)

16. Set data SW1 high, and move the clock input from logic switch A to the CLOCK connector on your Trainer. Set the CLOCK frequency to 1 Hz.
17. Move data SW1 back to a low. The cascaded counters are now switching at a rate of 1 Hz, or one count-per-second.
18. The counters will reach a binary output of 11111111 when all four logic indicators are lit, and the 7-segment LED displays an "F". (It will take 4:15 to reach this point.) When this point is reached, record the output voltage below. Be sure to read this voltage quickly; it will remain for only one second, then the counters will cycle down to a binary output of 00000000.

Output voltage (binary 11111111) = _____ V

Divide this voltage by 255 (the total number of steps) and record your result below:

_____ V

Is this voltage approximately equal to the original step voltage you calculated in Step 14?

_____ (yes/no)

19. Set data SW1 high, and increase the frequency of the CLOCK generator from 1 Hz to 100 kHz. Remove your voltmeter from the circuit and connect your oscilloscope to the output (pin 1 of the op amp).
20. Answer the following questions:
 - a. What is the voltage at the highest point on the waveform?

_____ V

- b. Is this value approximately equal to the value you measured in Step 18?

_____ (yes/no)

21. Calculate the frequency of the output waveform:

Output = _____ kHz

Is this frequency higher, lower, or equal to the frequency of the CLOCK generator?

(higher/lower/equal to)

Explain your answer:

22. Turn all equipment off, but leave the circuit connected. This circuit will be used in the Experiment 25.

Discussion

Steps 10 through 22 allowed you to view a DAC circuit in a more typical application. After wiring the circuit, you began by setting your input reference current, just as you did for the previous circuit.

Once the input reference current was set, you then had to adjust the output voltage to a set amount. In the previous circuit, you made this adjustment when the binary input was 1111, or after 15 steps. Adjusting the output voltage with an all-high input would be impractical for this circuit, however, because you would have to apply 255 clock pulses to reach an all-high input.

Another method you could have used to set your output voltage is to adjust the output with a binary input of 00000001, or after only one clock pulse. This would also be impractical, though. Remember, the circuit must go from 0 volts to approximately +12 volts in 255 steps. The voltage change for a single step would therefore be very small; too small for an accurate adjustment.

In Step 13, you adjusted the output voltage to +1.60 volts, after setting the binary inputs to the point where L3 was lit, and the 7-segment LED read "8". The binary equivalent to this input is 00101000, and its decimal equivalent is 40.

We can determine the amount of voltage change for each step with this information. First, we know we are using up-counters. Second, since the decimal input was 40, we know that the counter was pulsed 40 times, which also means that the DAC stepped 40 times. Third, we know that the output voltage after 40 steps is 1.60 volts. So the amount of voltage change for each step is equal to:

$$\frac{1.60 \text{ V}}{40 \text{ steps}} = 0.04 \text{ V/step}$$

You verified this value in Step 15, by randomly toggling logic switch A. You should have seen that for each toggle of logic switch A, the output voltage of the circuit increased by +0.04 volts.

The CLOCK generator of the Trainer was used for Steps 16 through 22. Starting with the 1 Hz setting in Step 18, you recorded the output voltage when the binary input was all high. Your value should have been very close to +10.22 volts. If your voltage was slightly different, remember that we are dealing with a step change of less than one-tenth of a volt—what you would have seen as +1.60 volts when you adjusted the output voltage could have easily been +1.605, which would throw off the stepping voltage, and the output in Step 18. When you divided your high voltage by 255, you probably reached a value very close to, but not exactly, 0.04 volts.

You then increased the frequency of the CLOCK generator to 100 kilohertz, and viewed the output with your scope. The highest voltage of the waveform should be very close to — if not equal to — the voltage you measured in Step 18.

Finally, you calculated the frequency of the output waveform, which should have been approximately 392 hertz. Although the clock frequency is 100 kilohertz, you still need 256 clock pulses to cycle the DAC from 0 volts back to 0 volts. Therefore, the frequency of the output waveform is 100 kHz divided by 256, or approximately 392 hertz.

ANALOG-TO-DIGITAL CONVERSION

The process of converting an analog voltage level into a binary number is known as analog-to-digital conversion. You will also hear the term “digitize” used in referring to the process of converting an analog input into a binary output number. The device that performs that function is called an analog-to-digital converter (ADC). A typical ADC has a single analog input and a multibit binary output. Depending upon the desired resolution of conversion, ADCs are available from 4 to 20 output bits. See Figure 10-21, which shows a general block diagram of an ADC.

A wide variety of different circuits have been devised to perform analog-to-digital conversion. Only a few of these techniques have found wide applications in electronics. Here, we discuss the most common methods of analog-to-digital conversion. Keep in mind that most of these devices are available as single integrated circuits. Like DACs, you can almost think of an ADC as a component and a simple building block of a more complex system.

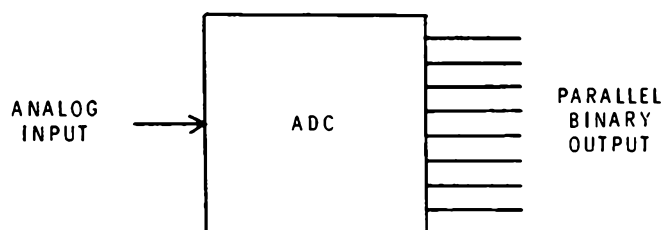


Figure 10-21

Simplified block diagram of an
analog-to-digital converter.

Comparators

Before we discuss the more common forms of ADC, it is important to consider one of the key elements in almost all ADC methods, the comparator. A comparator is a linear or analog circuit that compares two analog inputs and generates a single binary output.

Figure 10-22A shows a simplified symbol for a comparator. Note that it has two inputs and a single output. One input is designated the reference, and a reference voltage is usually applied to it. The input voltage is then compared to that reference. Any desired analog signal can be applied to this input. The comparator circuit looks at the two inputs and generates either a binary 1 or binary 0 output, based on the relative magnitudes of the inputs. For example, if the input voltage is below the reference value, the comparator output is binary 0. If the input voltage is higher than the reference voltage, the output will be binary 1. The simple diagram in Figure 10-22B shows the transfer function of the comparator. Note that the comparator switches at the point where the input and reference are equal.

Most comparators are very high gain differential amplifiers. They have a high input impedance to the analog input signal. Their output is usually limited or clamped to a standard binary (TTL) output level. Standard integrated circuit comparators are available.

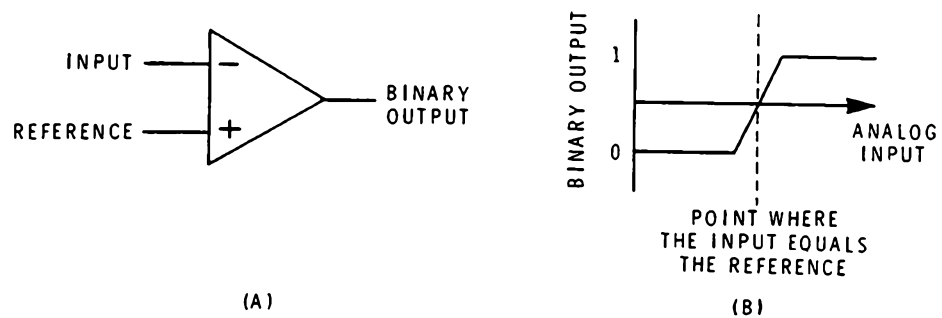


Figure 10-22

(A) Comparator symbol.

(B) Comparator transfer function.

A common integrated circuit op amp can also be used as a comparator. No feedback is used on the op amp (open loop configuration), allowing the amplifier to operate at its highest possible gain. The differential inputs of the op amp can be used for the reference and input signals, as shown in Figure 10-23A. Assume a +6 volt reference input. If the input is less than +6 volts, the output will be -0.7 volts, the forward biased drop across the zener. When the input first equals and then exceeds +6 volts, the comparator switches. The output at this time is +4.7 volts, the zener's reverse voltage.

An alternate method of applying inputs to the op amp used as a converter is shown in Figure 10-23B. Here, input currents are compared instead of voltages. The input voltages are converted to currents by the resistors at the input (S) of the op amp. Note that a zener diode is used in the feedback path to clamp the amplifier output to approximately the standard TTL levels of 4.7 and -0.7 volts.

The operation of the comparator is extremely simple. When the input voltage is below the reference value, the output is binary 0. When the input voltage is above the reference voltage, the output is binary 1. When the two inputs are equal, the converter switches. Because of the very high gain of the amplifier circuit, only a tiny voltage difference between the two will cause the comparator to switch from one state to the other. The smaller the voltage difference that causes switching, the better and more accurate the comparator. This voltage difference is referred to as the input offset voltage and is a measure of a comparator's sensitivity.

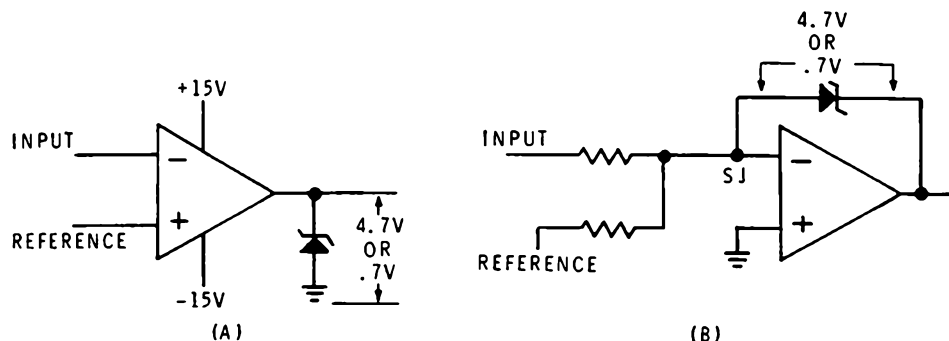


Figure 10-23

Using an op amp as a comparator.

(A) Differential inputs.

(B) Current inputs.

Counter-Ramp Feedback ADC

One of the simplest forms of ADC is the counter-ramp feedback circuit shown in Figure 10-24. The circuit consists of a binary counter, a DAC, a comparator, and some logic circuitry.

The analog signal to be converted into a binary value is applied to one input of the comparator. The other input to the comparator accepts the output from the DAC. The binary counter drives the DAC. The counter is incremented by a clock signal passed through NAND gate 1. Note that the output of the comparator also controls the NAND gate.

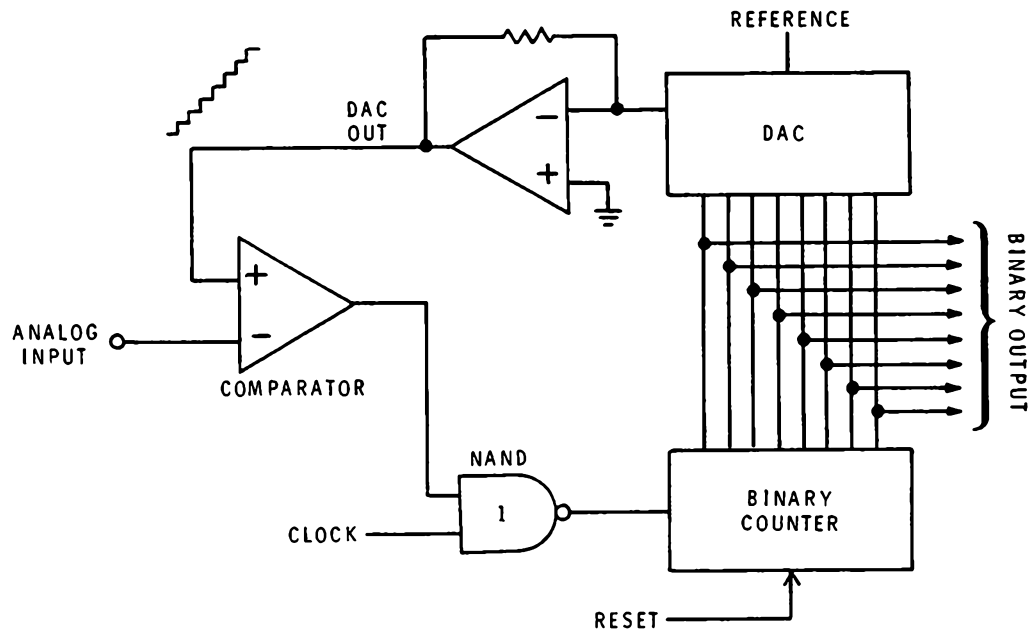


Figure 10-24

Counter-ramp feedback method
of A to D conversion.

CYCLE OF OPERATION

With the analog signal applied to the input, a reset pulse is applied to the binary counter. This start pulse resets the 8-bit binary counter so that it begins counting from 0.

With the binary counter at 0, the DAC output will also be 0. At this time, the analog input signal is greater than the DAC output. Therefore, the comparator output is a binary 1 and the NAND gate is enabled, allowing the clock pulses to increment the binary counter.

As the binary counter is incremented, the DAC generates a stair-step output voltage. As soon as the DAC output equals or surpasses the analog input signal level, the comparator output will switch to binary 0, thus, inhibiting the clock pulses. At this time, the conversion halts and the binary counter output is the digital equivalent of the analog input. The binary output value is proportional to the analog input signal, depending upon the reference voltage applied to the DAC and the op amp scaling factor.

There are several important things to point out about this particular type of ADC. The most important is that the conversion time is proportional to the analog input value. The higher the analog input value, the longer the counter is incremented until the DAC output rises to meet the analog input. Small input voltages produce short conversion times, while high input voltages produce long conversion times. See Figure 10-25.

Conversion time also depends on the clock frequency. The higher the clock frequency, the faster the conversion. Finally, the number of bits in the counter affects the conversion time. The maximum conversion time is $2^N - 1$ (N = number of bits) multiplied by the clock period.

Further, since conversion time depends on the amplitude of the analog input, conversion times are unequal for different input values. This is sometimes a disadvantage. It is more desirable to have a fixed conversion time regardless of the analog input value.

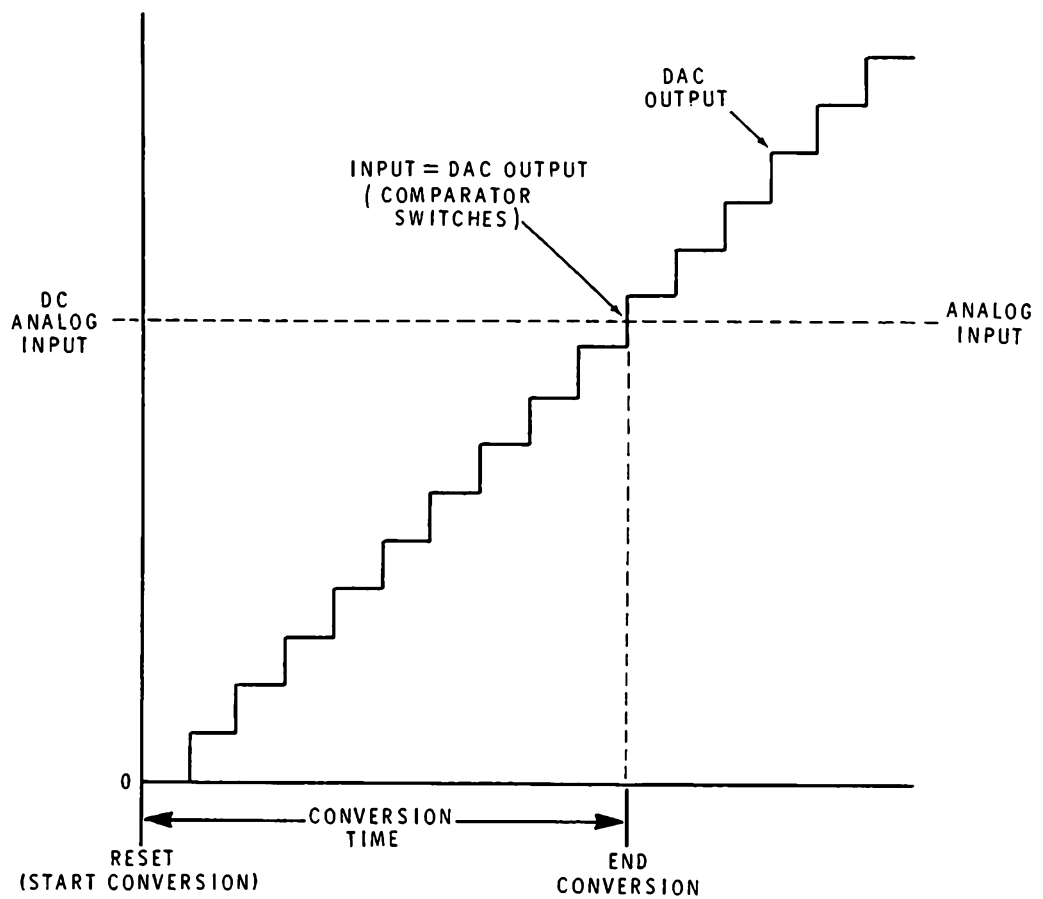


Figure 10-25

Operation of a counter feedback ADC
showing conversion time.

Successive Approximations Converter

An improved version of the counter-ramp feedback converter is known as the successive approximations converter. This form of ADC is significantly faster than the counter-ramp method and is by far the most popular form of ADC. Figure 10-26 shows a general block diagram of a successive approximations converter. Like the counter-ramp circuit, the circuit contains a DAC and a comparator. Instead of a binary up-counter driving the DAC, a special successive approximations register (SAR) is used. This special sequential circuit contains flip flops and other logic that supply a special sequence of binary numbers to the DAC. A clock pulse steps the flip flops in a unique sequence.

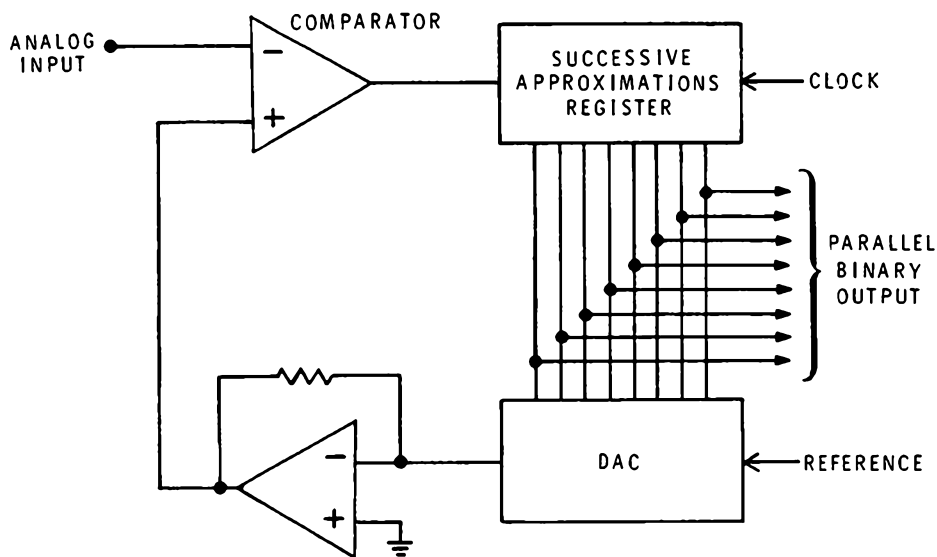


Figure 10-26

Successive approximation ADC.

Figure 10-27 shows the sequence of binary numbers applied to the DAC when a conversion is initiated. These are the various decisions which are made by the comparator as the bits in the SAR change. For simplicity in this example, a 4-bit SAR and DAC are assumed.

When a conversion is initiated, the 4-bit SAR is initially set to 0. Immediately, the most significant bit is set. The MSB causes the DAC output to be one-half of the input reference voltage. This is applied to the comparator along with the analog input. If the analog input is less than the DAC output, the most significant bit is turned off and the next most significant bit is turned on. If the analog input voltage is above the DAC output, the MSB remains set and the next most significant bit is set. Again, the comparator looks at the input signal, the DAC output, and generates an output that tells the relationship of the two. The bits in the SAR are continually set and reset, from MSB to LSB, depending upon the comparator output. After each comparison, the next most significant bit is set or reset. This process continues until the final binary output is developed.

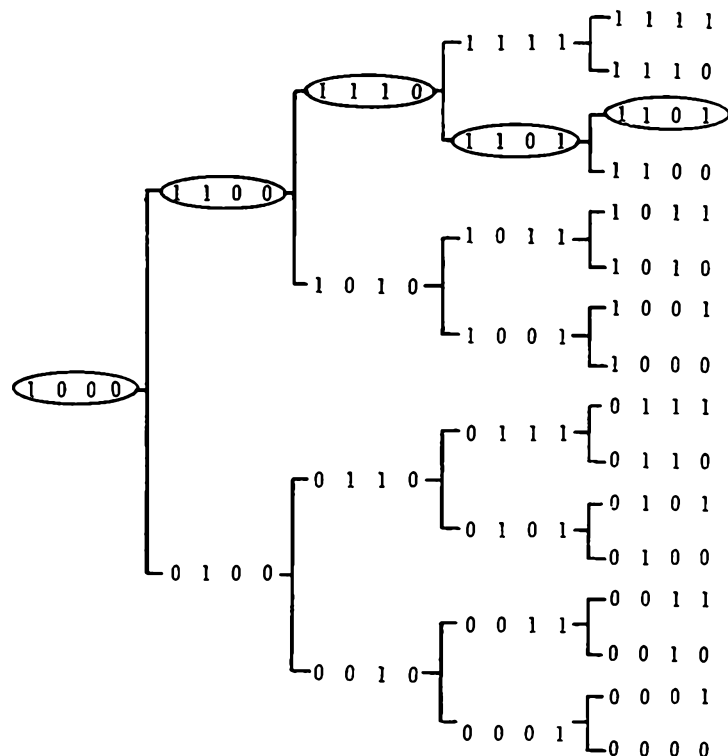


Figure 10-27

Possible number sequences of a 4-bit successive approximations ADC.

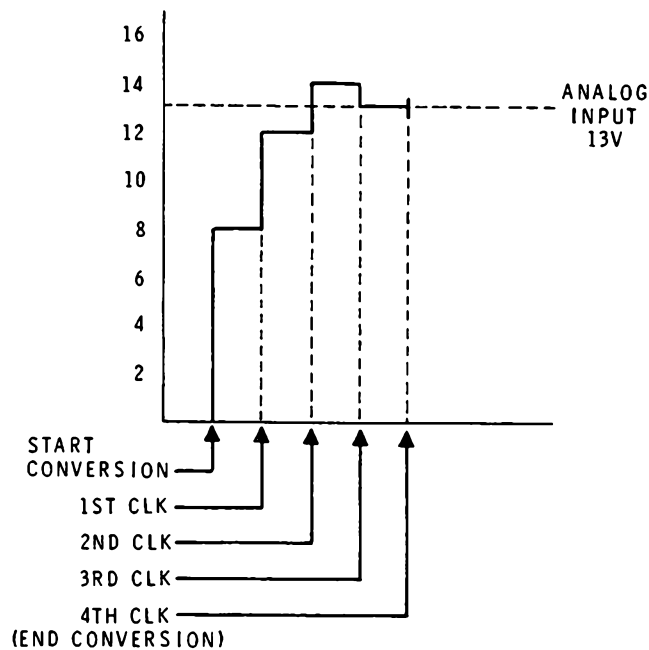


Figure 10-28

DAC output in successive
approximation ADC.

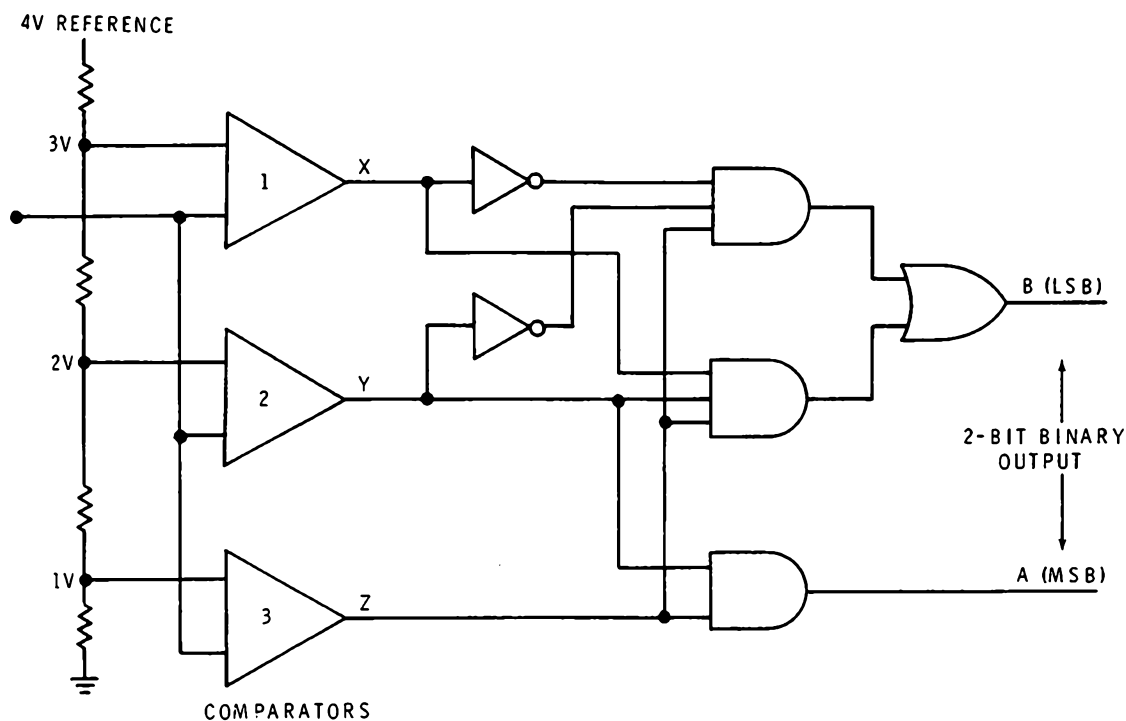
Assume a 13 volt input to a 4-bit ADC. Figure 10-28 shows the DAC output as each comparison decision is made. When the conversion begins, the MSB is turned on, generating an 8 volt output. This is less than 13, so the comparator output signals the SAR to set the next bit. The next most significant bit is set and the output 12 occurs. Again, this is less than the input, so the next most significant bit is set, making the output 14. The DAC output is now greater than the input, and the SAR turns off the last bit set and turns on the LSB. This makes the DAC output equal to the input and the conversion is complete. Note this path on Figure 10-27.

The primary advantage of the successive approximation conversion technique is its high speed. Remember, the counter-ramp converter's conversion speed was proportional to the number of output bits. The conversion time in a successive approximation ADC is N times the clock period. Only N comparisons (N = number of bits) or decisions are required for each conversion. Conversion times of less than 2 microseconds for word lengths up to 16 bits are possible with modern IC ADCs of this type.

Flash Converters

One of the simplest and most direct methods of analog-to-digital conversion is accomplished by the flash converter. Also known as the parallel or simultaneous method of A-to-D conversion, this method is the fastest of all A-to-D conversion processes.

The principle operation of a flash converter is to compare the analog input signal to be digitized simultaneously to successively smaller, equally spaced increments of a reference voltage. The outputs of the comparators used in this process will then indicate the voltage level of the analog input. The comparator outputs are then converted into a binary code proportional to the analog input value. The speed of conversion is limited only by the switching speed of the comparators and the relatively short propagation delay of the logic network used to generate the binary code. Conversion times of only a few nanoseconds are possible with flash conversion.



INPUT (V)	COMPARATOR OUTPUTS			OUTPUT	
	X	Y	Z	A	B
0 - 1	0	0	0	0	0
1 - 2	0	0	1	0	1
2 - 3	0	1	1	1	0
3 +	1	1	1	1	1

Figure 10-29

A flash converter.

To illustrate this concept, consider the simple two-bit flash converter shown in Figure 10-29. The number of bits refers to the size of the parallel binary output word. This, of course, indicates the resolution of the conversion. With two bits, it is possible to define four possible states represented by the binary codes: 00, 01, 10, and 11. This means that the resolution is effectively 1 in 4.

To achieve this resolution, three comparators are used. The analog input signal is applied simultaneously to one input of each comparator. The other input to each comparator is derived from a voltage divider network driven by a precision reference voltage. Note that the voltage divider supplies voltages of 1, 2, and 3 volts to the comparators. Assume

that the analog input voltage will vary over the zero to 4 volt range. Further assume that when the analog input voltage is less than the voltage applied to the other comparator input, the comparator output will be binary 0. When the analog input voltage exceeds the reference voltage at the converter input, the comparator will generate a binary 1 output. With this arrangement you can see in Figure 10-29 the binary state of each comparator with any input voltage.

The comparator output voltages are then fed to a combinational logic network to perform the function of a code converter. It converts the comparator outputs into a two-bit binary number that indicates the value of the analog input. You can check this relationship for yourself by assuming some value of analog input voltage between 0 and 4 volts, and then determining the comparator outputs and the resulting binary output.

The problem with this simple two-bit converter is that its resolution is low. It does not provide a precise binary representation of the analog input. For example, a 2.5 volt input would simply be indicated by a 10 (2) output. The binary output is certainly a close approximation of the analog input, but for most applications higher resolution is required. This is accomplished by using a greater number of input comparators. The number of input comparators (C) required to achieve N bit resolution is expressed by the equation below.

$$C = 2^N - 1$$

For example, to achieve a 4 bit or 1-in-16 resolution, 15 comparators are required. Six-bit resolution is achieved with 63 comparators. This is a considerable amount of circuitry, and the resulting combinational code conversion network becomes increasingly complex. The problem becomes even more difficult when you attempt to integrate all of this circuitry on a single silicon chip. Nevertheless, fully integrated flash converter ICs are available. Four- and six-bit devices are commonly available. Eight-bit resolution flash A-to-D converters have been built in integrated circuit form. With 255 comparators and a highly complex binary conversion network, they represent the state-of-the-art in combined linear-digital integrated circuitry. Conversion speeds of less than 10 nanoseconds are achievable with such devices.

Flash A-to-D converters are used in applications requiring the digitizing of very high frequency analog signals. A common application is in digitizing video signals into binary numbers that can be processed by digital computers.

Dual Slope ADC

The dual slope method of analog-to-digital conversion is one of the most accurate methods available. Because of its high accuracy, this method is widely used in instrumentation where precision measurements of analog signals is the primary application.

The dual slope method is based on the principle of converting an unknown analog input voltage to a time interval that can be measured with a digital counter. A general block diagram of a dual slope converter is shown in Figure 10-30. The key circuits are an integrator, a comparator, and a binary or BCD counter. In most applications, a BCD counter is used with direct decimal readout on seven segment LED or LCD displays.

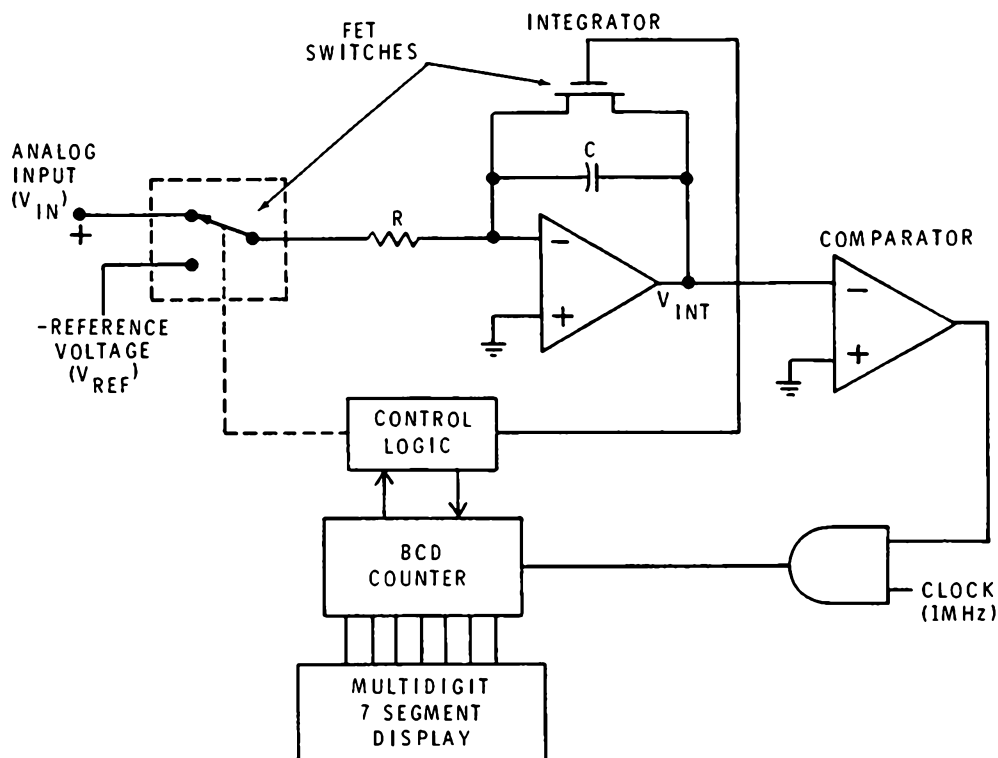


Figure 10-30

Dual slope ADC.

Note in Figure 10-30 that the input to the integrator can be switched between two separate sources, the unknown analog input voltage, and a precision reference voltage. The principle of this converter is to allow the integrator to integrate the unknown input voltage for a fixed period of time. At the end of that time period, the output voltage of the integrator is a function of the unknown input voltage. The integrator input is then switched to the precision reference voltage. The integrator integrates this voltage until the output of the integrator is zero. During this time period, a known clock increments the counter until the integrator output is zero. The time that it takes for the integrator to reach zero depends upon the value of the reference voltage and the output voltage of the integrator which, in turn, was determined by the value of the unknown input. That time interval is proportional to the value of the unknown input voltage.

In order for you to understand the operation of this circuit, it is important that you understand how an integrator works. The integrator in this application is an op amp with an input resistor and a feedback capacitor. The output of the integrator is the time integral of the input voltage. To understand the operation of the circuit, it is not necessary to go into the complex mathematical relationships involved. It is simply necessary to know that the output of the integrator is simply a ramp, or sawtooth, when the input is a DC voltage. For example, with a positive DC input voltage, the feedback capacitor is charged by a constant current equal to the value of the input voltage divided by the input resistance. This constant current charges the capacitor linearly. The integrator output, therefore, is a negative-going ramp. The rate of rise of the ramp depends upon the RC time constant of the integrator resistor and capacitor as well as the value of the input voltage. The integrator output voltage (V_{int}) can be expressed by the simple equation below.

$$V_{\text{int}} = -(V_{\text{in}} \times T_1)/RC$$

T_1 = The fixed time period that the integrator is allowed to change to the input reference voltage.

As you can see, the integrator output is proportional to the value of the input voltage and inversely proportional to the RC time constant. The output voltage continues to rise in a straight line as long as the constant input voltage is applied.

One last point to note is that the integrator is an inverter. If the input voltage is positive, the output voltage is negative. And vice versa. A positive-going voltage produces a negative-going output, while a negative-going input produces a positive-going output. Now with this information in mind, let's describe the dual slope method in more detail.

A conversion cycle begins when the control logic circuit resets the counter and discharges the integrator capacitor with a FET switch. The integrator output is zero and the input of the integrator is simultaneously switched to the unknown analog input voltage to be digitized. The input switching is accomplished by FET switches (not shown). This causes the comparator output to enable the AND gate so that 1 MHz clock pulses are applied to the counter.

While the clock pulses begin incrementing the counter, the integrator begins to linearly charge its feedback capacitor at a rate dependent upon the value of the analog input signal. The counter continues to count until its maximum count value is reached. At this time, one additional clock pulse will recycle the counter to zero. For a three-digit counter, the counter counts from 0 through 999. This represents 1000 counts, or 1000 input clock pulses. When the counter recycles to zero, the zero state is detected by the control logic. This switches the integrator input from the unknown analog input voltage to the precision reference voltage. The polarity of the reference voltage is always opposite that of the polarity of the input signal. This means that, with a positive input, a negative reference voltage will now cause the integrator output to go in a positive direction.

Up to this point, what we have done is to allow the unknown analog signal to generate an output ramp voltage for a fixed period of time. That period of time is designated T_1 and is equal to the maximum count capability of the counter multiplied by the period of the clock pulses. In this case, with clock pulses occurring every 1 microsecond and a count capability of 1000, we have allowed the integrator to integrate the input voltage for a period of 1 millisecond. At the end of this period of time, the output voltage of the integrator is proportional to the unknown analog input voltage. Figure 10-31 shows the negative ramp voltage generated by the positive unknown input signal during T_1 .

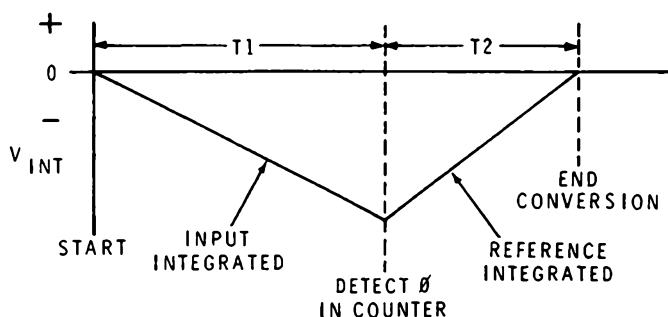


Figure 10-31

Integrator output in a
dual slope ADC.

The integrator now begins to integrate the negative reference voltage. For that reason, the integrator output begins to move in a positive direction. The counter begins counting up. When the integrator output passes through zero, the comparator output switches from binary 1 to binary 0. Note that with one input to the comparator at ground, the comparator will switch from one state to another when its input is above or below zero volts. When the comparator output goes to binary 0, it inhibits the AND gate and stops the clock pulses from reaching the counter. At this time, the number in the counter is proportional to the unknown analog input voltage. With everything properly adjusted, the display connected to the counter will show the actual value of the unknown input voltage.

During the time that the integrator integrates the reference voltage, the output is:

$$V_{\text{int}} = (V_{\text{ref}} \times T_2)/RC$$

Here, T_2 is the time it takes for the reference voltage to discharge the feedback capacitor from the maximum value obtained by integrating the input. This is expressed mathematically as $(V_{\text{in}} \times T_1)/RC - (V_{\text{ref}} \times T_2)/RC$. Reducing this algebraically gives

$$V_{\text{in}} = V_{\text{ref}} \times T_2/T_1$$

The unknown analog input V_{in} is expressed as the ratio of T_2/T_1 times the reference voltage.

To summarize, what the conversion cycle accomplished was to integrate an unknown input voltage for a fixed period of time. The output of the integrator at the end of that period of time was a function of the unknown input voltage. The integrator was then switched so that it integrated a known reference voltage. The time it took and the number of counts required for the integrator output to return to zero was directly proportional to the input voltage. By properly sizing the reference voltage, the BCD number stored in the counter was equal to the unknown input voltage.

The dual slope method of analog-to-digital conversion has some very important advantages over other methods. First, is its accuracy. The accuracy of the measurement of the dual slope converter is primarily dependent upon the quality of the reference voltage. Virtually none of the other characteristics of the circuit have any degree of influence over the accuracy. This includes the absolute value and temperature stability of the integrator RC time constant, the clock frequency, or the comparator switching speed. Even if the clock frequency and RC values change over the long term, this does not affect the measurement accuracy. The accuracy is dependent upon the stability of those values during the period of integration. As you have seen, this is only several milliseconds in length and, therefore, little change can take place over a short period of time.

Another advantage of this circuit is its inherent noise rejection. Since the integrator is essentially a low-pass filter, it has the effect of removing any high frequency variations or short noise bursts that might be present at the input. Low cost is another major advantage of this circuit. The circuitry is basically noncritical and, therefore, standard components can be used to implement it. Today, however, entire dual slope converters are available on a single IC chip.

The main disadvantage of the dual slope method is its slow conversion time. Because it must integrate the input voltage and the reference voltage, typical conversion times are in the 1 millisecond to 1 second range. A total conversion time of 10 to 100 milliseconds is typical for most units. This is many orders of magnitude slower than the previously discussed methods of conversion. Nevertheless, despite the slow speed, it is fast enough for many applications.

The primary application for dual slope converters is in digital multimeters. These widely used instruments are capable of measuring unknown voltages and conveniently displaying them in decimal form on their readouts. Additional input circuitry allows both AC and DC voltages and currents as well as resistance to be measured.

Voltage-to-Frequency Conversion

One of the lowest cost forms of analog-to-digital conversion is a technique referred to as voltage-to-frequency (V-to-F) conversion. This is a process where a DC voltage or varying analog signal is converted into a repetitive pulse train. The output is a series of fixed width digital pulses whose repetition rate (frequency) varies in proportion to the input voltage. For a low input voltage, a low output frequency is produced. As the input voltage increases, the pulse repetition rate increases proportionally.

One method of implementing a V-to-F converter is shown in Figure 10-32. Notice that the input is a DC voltage which is applied to an op amp integrator. The output of the integrator is applied to one input of a comparator whose reference input is a fixed DC voltage. The comparator drives a bipolar transistor switch connected across the integrator capacitor. This serves to discharge the capacitor when the integrator output equals the reference voltage level. The comparator also drives a one-shot multivibrator, which generates fixed amplitude constant duration digital output pulses.

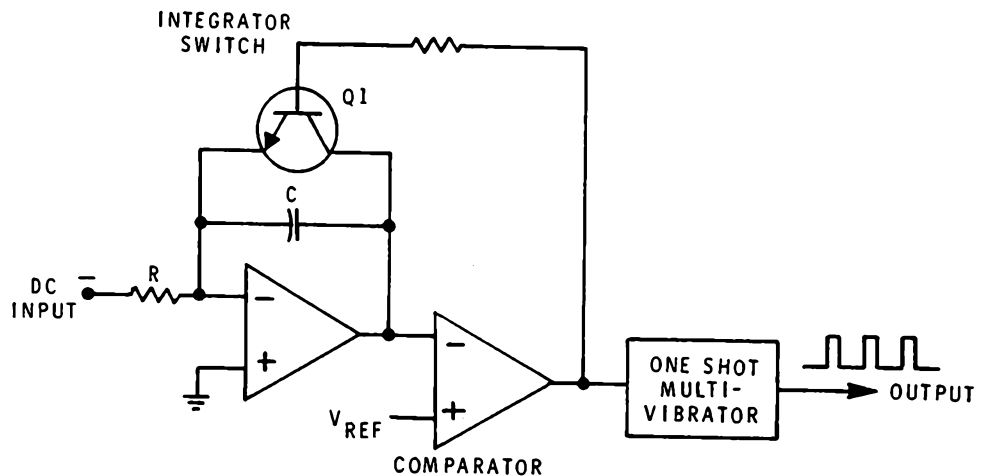


Figure 10-32

A voltage-to-frequency converter.

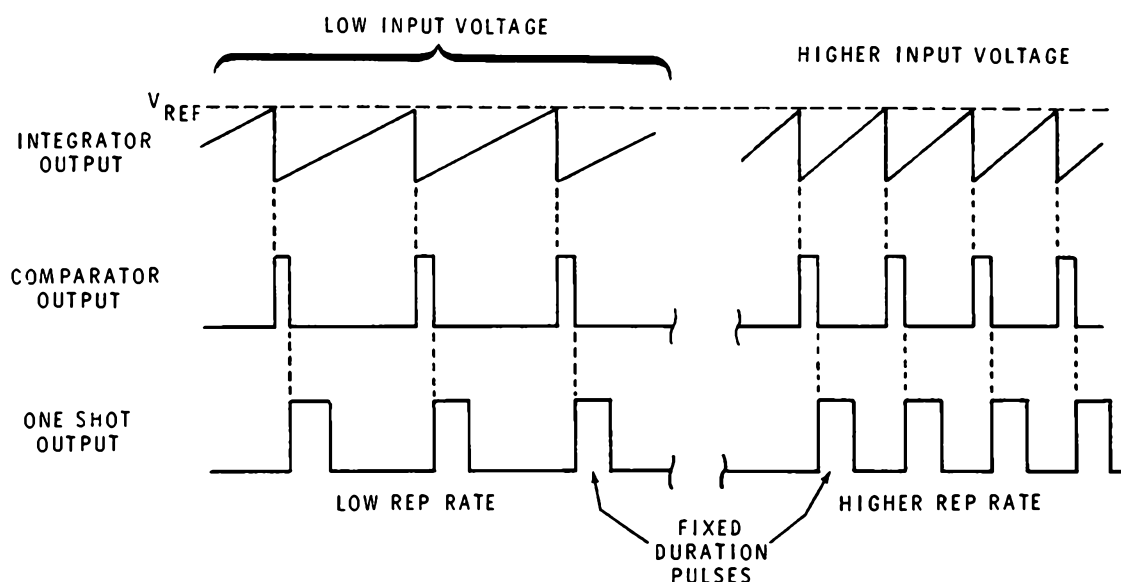


Figure 10-33

Operating waveforms of a
voltage-to-frequency converter.

The waveforms in Figure 10-33 show how the V-to-F converter works. With a fixed DC input voltage, the integrator capacitor charges at a linear rate, generating a linear output voltage ramp. When the output voltage equals and then exceeds the value of the comparator reference voltage, the comparator triggers and turns on Q1, which discharges the integrator capacitor. The comparator also triggers the one shot, which generates an output pulse. When Q1 discharges the integrator capacitor, the integrator output drops below the level where the comparator triggers and the cycle repeats itself.

The higher the input voltage of the circuit, the faster the capacitor charges. As the rate of charging increases, the comparator threshold voltage is reached more quickly; therefore, the one-shot is triggered more often. Increasing the voltage increases the rate of charge and the frequency with which the one-shot output pulses are generated. As you can see, the one-shot output is a series of pulses whose repetition rate is proportional to the input voltage.

In most applications, the output of a V-to-F converter is not as convenient to use as is the parallel digital output of other ADCs. As a result, additional circuitry is usually employed to convert the varying frequency output pulse train of the V-to-F converter into a parallel binary word. This is done with a simple counter circuit. The counter circuit is used to measure the frequency of the output pulses and display it as a binary number. This is done by counting the number of output pulses that occur over a fixed interval of time.

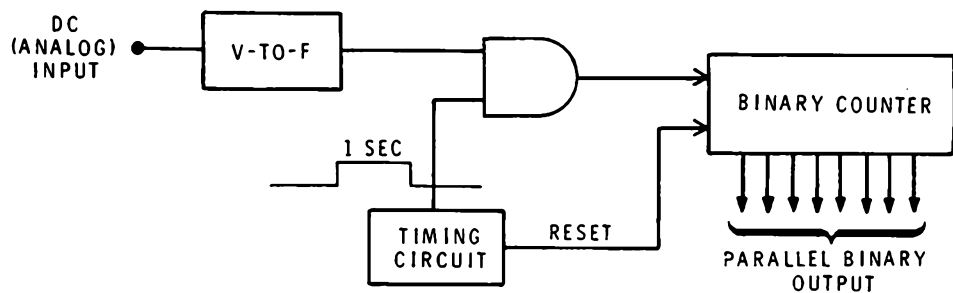


Figure 10-34

Converting the V-to-F output
into a parallel binary word.

A circuit for doing this is shown in Figure 10-34. Here a precision timing circuit generates a fixed duration pulse one second wide. This is applied to one input of the AND gate. The output of the voltage-to-frequency converter is applied to the other input of the AND gate. The counter is first reset. Then, during the one second interval the AND gate is enabled, the counter is incremented by the V-to-F converter output pulses. Counting the number of pulses that occur in one second is the process of measuring the frequency. The binary number contained in the counter represents the frequency in pulses or cycles per second. If the counter has eight bits and at the end of one second the counter contains the number 175, the output frequency of the V-to-F converter is 175 Hz.

The voltage-to-frequency converter is used in digital multimeters and digital panel meters. It is also used in instrumentation systems where various analog quantities must be measured and transmitted to be stored or displayed in some remote site. Transducers for measuring light, temperature, pressure, and other physical quantities are designed to generate linear output voltages. These can be applied to V-to-F converters in order to translate the analog quantity into a digital signal. Pulses from the V-to-F converter are then transmitted over a transmission line and reconstructed at the remote site, where their frequency is measured with a counter arrangement like that described earlier.

While V-to-F converters are the simplest and lowest cost form of A-to-D conversion, they are also the slowest form of ADC. Conversion times from 50 milliseconds to as long as several seconds are typical for such devices. However, in the applications where these converters are normally used, this slow conversion time is not a disadvantage.

ADC Specifications

Like DACs, ADCs have specifications that indicate the performance level or quality of the converter. The previously covered DAC specifications, such as resolution and linearity, also apply to the ADC. However, there are some additional ADC specifications which are important. These are quantizing error, aperture error, and conversion time.

QUANTIZING ERROR

All ADCs have quantizing error. This is the error that occurs because ADC divides the analog input signal range into a number of discrete voltage increments. Because of this, quantizing error is directly related to resolution. Resolution, in turn, is dependent upon the number of bits in the digital output word. With an 8-bit word, the ADC divides the maximum input signal range into 256 discrete levels. As a result, the ADC cannot recognize input voltage level changes that are less than the voltage change produced by the LSB. For example, with an 8-bit data word and a 10 volt reference, the minimum voltage increment or step is 39 millivolts. This ADC can not recognize input voltage changes less than this value.

Since quantizing error results from limitations in resolution, it is usually specified as a fraction of an LSB change or a percentage. When expressed as a percentage, quantizing error is usually referenced to the full scale voltage input. An ADCs basic quantizing error, then, is one LSB voltage change and it cannot be reduced for a given converter. The LSB change is usually expressed as $\pm 1/2$ LSB.

APERTURE ERROR

Aperture error is the voltage change that occurs in an analog signal that varies during the conversion period. Since it takes a finite period of time for an ADC to develop a digital output for a given input voltage, it is possible that the analog signal could change in that period of time. If the change is more rapid than the conversion of time, a considerable conversion error can exist. This is referred to as aperture error.

For most applications where the input is a DC voltage or a slowly varying AC signal, the aperture error is negligible. But when high speed analog input signals are used, whose frequency or changes are of the same order of magnitude as the conversion time, aperture errors can be considerable. In the next section we will show a method that virtually eliminates aperture errors.

CONVERSION TIME

Conversion time is the interval required for an ADC to change the analog signal into a digital output. As you saw with the various types of ADCs, it takes a finite period of time to perform this conversion. Counter-ramp and dual ramp converters have relatively long conversion times. Successive approximation converters are faster. Flash converters are the fastest, but even they require a certain amount of time for the conversion to take place. The conversion time is typically expressed in fractions of a second. For the flash converter, conversion times are in the nanosecond region. Successive approximation converters can perform conversions in less than 10 microseconds. Other forms of converters typically take many hundreds of microseconds, milliseconds, or even longer in some cases.

Sampling and Multiplexing

So far in our discussion of ADCs, we have pretty much assumed that the analog input is a fixed or slowly varying DC voltage. While this is probably the case in a high percentage of applications, in other applications the analog input will be a varying DC or in some cases a high frequency AC signal. In addition, there are some applications in which the ADC must be able to convert the inputs from multiple analog sources to digital numbers. With some additional circuitry, these special applications can be accommodated.

SAMPLING

Whenever an ADC converts a fixed DC input voltage into its binary equivalent, the process is a relatively fast and simple one. This binary output number can be stored or processed as required by the application. As long as the DC input remains constant, the output binary number will remain the same. However, how does the ADC deal with a varying DC or rapidly changing AC input? In most applications, the ADC is repeatedly sampling the input analog signal and generating a proportional binary output number. In the case of a DC input, the output is simply a series of binary numbers that are all the same. However, when the input is varying in amplitude, the output of the ADC will be a sequence of parallel binary numbers representing the analog voltage values at the instant the ADC samples the input and makes its conversion.

As an example, consider the sine wave shown in Figure 10-35. Assume that a successive approximation converter is used to convert the analog input into a sequence of output binary numbers. Assume further that the ADC conversion speed is 10 microseconds. Thus, every 10 microseconds, the ADC will sample the analog input voltage. Because the sine wave input is changing, the ADC sees a different input voltage value each time it makes a conversion. In this example, we show the ADC making 10 samples or conversions per cycle of the analog input. Therefore, one cycle of the analog input signal is represented by a sequence of ten 6-bit binary numbers appearing every 10 microseconds at the output of the ADC. These numbers may be stored in memory for future processing or be sequentially processed and displayed as the application calls for.

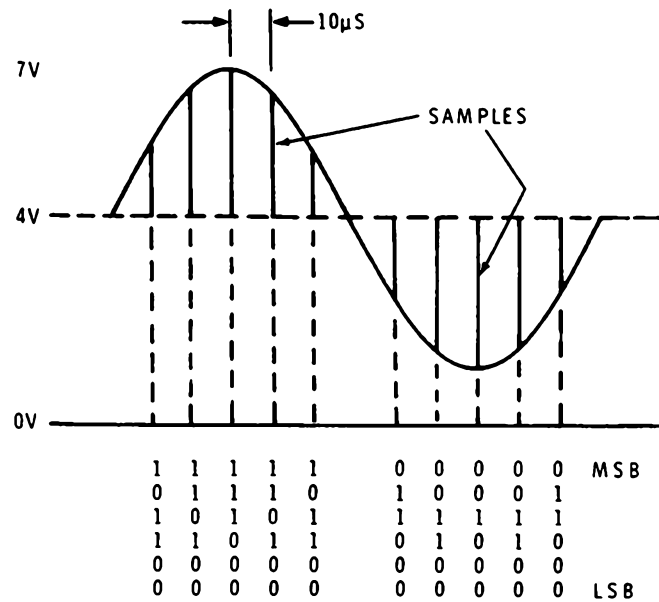


Figure 10-35

ADC sampling of a sine wave
(ten times per cycle).

One of the most important factors to consider in converting an AC analog signal into its digital equivalent is how many times per cycle the AC signal is sampled by the ADC. If the signal is sampled many times during the cycle, as in Figure 10-35, a fairly accurate digital representation of the analog signal will appear at the ADC output. The fewer the samples per cycle used, the poorer the representation. If too few samples per cycle are used, the digital output will bear little or no resemblance to the analog input. An absolute minimum number of samples per cycle for most applications is two. Figure 10-36 shows how this minimum of two samples per cycle might appear in an actual case. Here, the ADC samples the analog signal once on the positive cycle and once during the negative cycle. Any fewer samples will result not only in considerable error but also an output signal that is not at all related to the input.

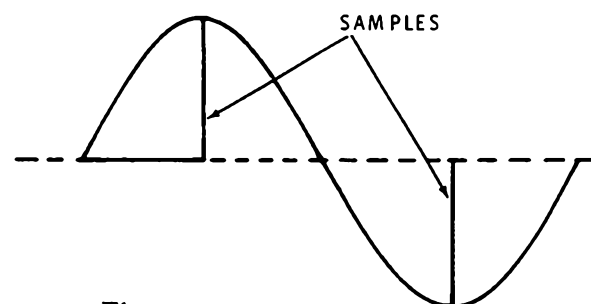


Figure 10-36

Sampling two times per cycle.

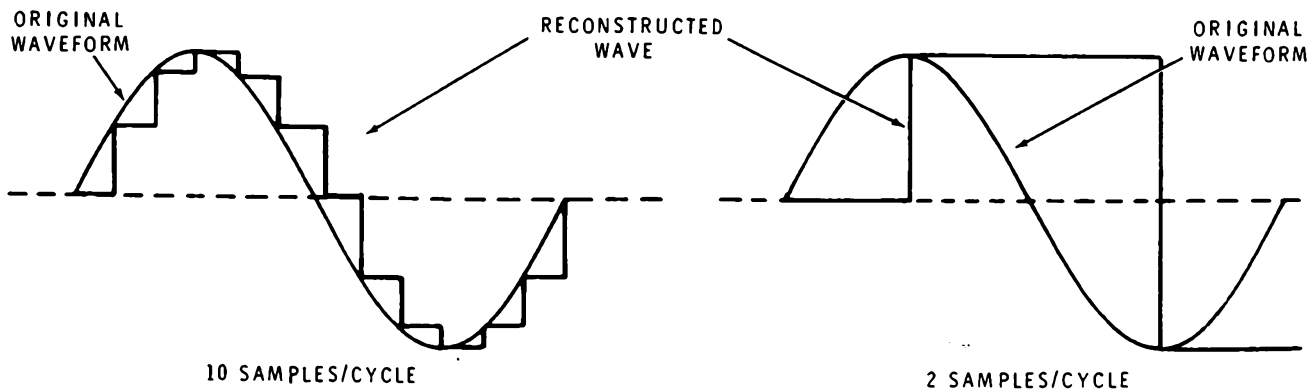


Figure 10-37

Reconstructing a digitized waveform with a DAC.

One way to better see this is to assume that the binary numbers derived from sampling an analog signal are stored away in a digital memory. The contents of this digital memory is then fed to an ADC, which reconstructs the analog signal. Figure 10-37 shows the results of a DAC converting the binary numbers representing an analog signal sampled 10 times per second and 2 times per second. Notice that the signal that was sampled 10 times per second is reconstructed in such a way that it is a reasonable-representation of the original analog signal. However, with only two samples per cycle, the analog output bears only the remotest equivalency to the original input signal.

The general rules to follow in determining the number of times an analog signal should be sampled are relatively simple. First, there should be no fewer than two samples per cycle in any application. This is the absolute minimum number and probably should never be used. On the other hand, the more samples per cycle used the better the representation of the input. As a general rule, 10 samples per cycle provides an optimum value. More samples per cycle will not produce much better representation of the original input. Fewer than 10 samples will, of course, degrade the signal when it is reconstructed by a DAC.

Using these guidelines, you can then determine just how fast the ADC must be in order to sample the desired analog signal. You must know the frequency of the analog input signal so that you can determine its period and hence the desired or required speed of conversion.

For example, assume the input is a 120 Hz sine wave. The minimum sampling frequency is twice the input, or 240 Hz. An optimum sampling frequency is ten times the input, or 1200 Hz. The ADC sampling rate (t) then is:

$$t = 1/f = 1/1200 = .0008333 \text{ second or } 833.3 \text{ microseconds}$$

As you may suspect, the higher the frequency of the analog signal, the faster the required conversion speed to maintain a high quality output. When you are dealing with complex analog signals made up of a wide range of frequencies, the sampling frequency is harder to determine. For example, if you are dealing with a voice input whose frequency range is approximately 300 to 3300 Hz, it is more difficult to choose a sampling frequency. You could use 10 samples per cycle on the highest frequency, giving a sampling frequency of 33 kHz. Or, you could use 10 samples per cycles of the middle frequency of the given range. For example, the audio frequency range is 3300 - 300 or 3000 Hz. The center frequency is one-half of this, or 1500 Hz. Ten samples per cycle would then require a conversion speed of 15 kHz, or a 66.67 microsecond sampling rate. This is just a general rule of thumb, and other rates will work.

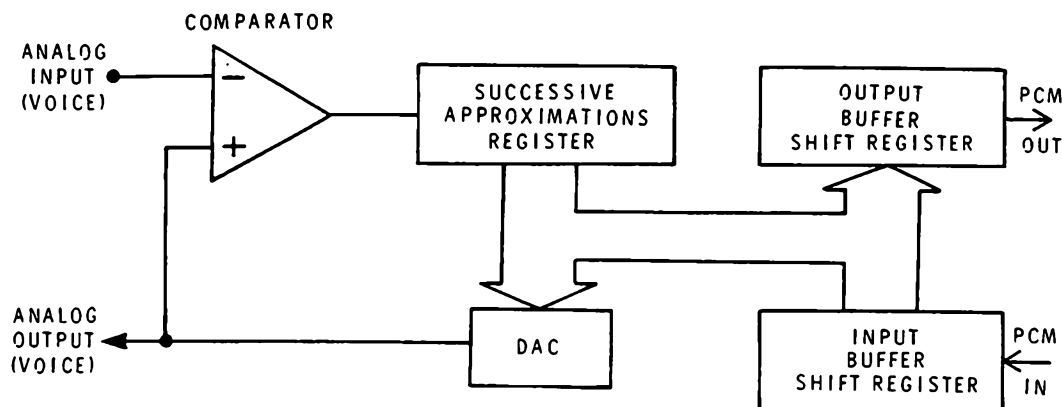


Figure 10-38

General block diagram of a CODEC IC.

CODEC

For example, consider an application involving CODEC, a special integrated circuit used for converting analog voice signals into digital signals and back again. A general block diagram of a CODEC is shown in Figure 10-38. The analog input signal to the device is usually a voice signal in the 300 Hz to 3.3 kHz range. This signal is applied to a successive approximation ADC which converts the analog input signal into an 8-bit digital word. This CODEC samples the analog input signal at an 8 kHz rate and generates a series of 8-bit binary words for each sample of the analog input. In this application it has been determined that a sampling rate of only 8 kHz is adequate for the desired performance.

These samples are fed to an output buffer (shift register), which then shifts the binary words out serially. The result is a pulse train of serial data words that represent the analog input signal. This form of output is generally referred to as pulse code modulation (PCM). This serial-binary output waveform is then used to modulate an analog carrier that is sent by telephone line or radio to some remote site.

Notice in Figure 10-38 that the CODEC not only generates PCM outputs but also accepts PCM input. This means that a PCM signal can be applied to the CODEC and the CODEC will perform the necessary serial-to-parallel and digital-to-analog conversion. The PCM input buffer is a shift register which accepts the serial binary input words and feeds them in parallel to the DAC. The DAC changes the signal into the analog equivalent. The CODEC reconstructs the digital input back into the original analog voice signal.

CODECs are designed primarily for use in digital telephone systems. Most telephone systems today use digital switching to take full advantage of computer control. CODECs can be used in any application where it is desirable to digitize voice signals for transmission from one point to another. PCM is ideal for the accurate transmission of voice information over various communication links where high noise environments must be tolerated.

SAMPLE AND HOLD

We have already described several times a special condition that might occur when an ADC is converting an AC analog input signal. That is the case where the input voltage varies during the time the converter is performing its conversion. This creates a serious error and the resulting output is meaningless. What is needed is a circuit that will "freeze" the analog input signal for an instant of time to allow the ADC to perform its conversion. A circuit which performs this function is called a "sample and hold" (S/H) circuit, or S/H amplifier.

An S/H circuit is an analog memory. Also called a track/store circuit, the S/H amplifier is an analog device controlled by a digital signal. In its normal operating state, the circuit samples or tracks the analog input. In other words, its output is the same as the input. When a binary 1 is applied to the digital input of the S/H circuit, the circuit holds or "freezes" the output at the analog value the instant the binary input is applied. It holds the analog output at that level.

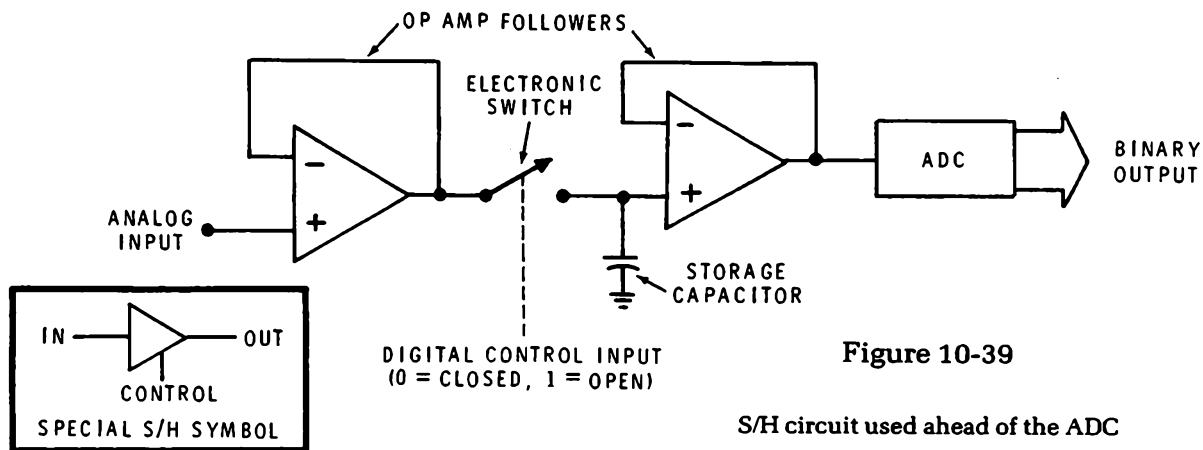


Figure 10-39 shows a diagram of a typical S/H circuit. It consists of an input op amp follower, an electronic switch, a capacitor, and an output op amp follower.

Recall that an op amp follower has a very high input impedance and, therefore, causes little or no loading of an analog signal. Its output impedance is extremely low and can drive a considerable load; even capacitive loads can be utilized. Op amp voltage followers have unity gain and no inversion.

Refer to Figures 10-39 and 10-40. The analog input sine wave is applied to the first op amp. Its output is connected to an electronic switch. This switch can be a FET, bipolar transistors, or a diode bridge. The switch is turned off and on with a digital control signal (binary 0 = on or closed, binary 1 = off or open). When the switch is on, the analog signal is applied to the capacitor and the input of the second op amp. The input follower charges and discharges the capacitor in accordance with the input. The output of the second follower is simply an identical buffered version of the original input. The digital control input at this time is binary 0.

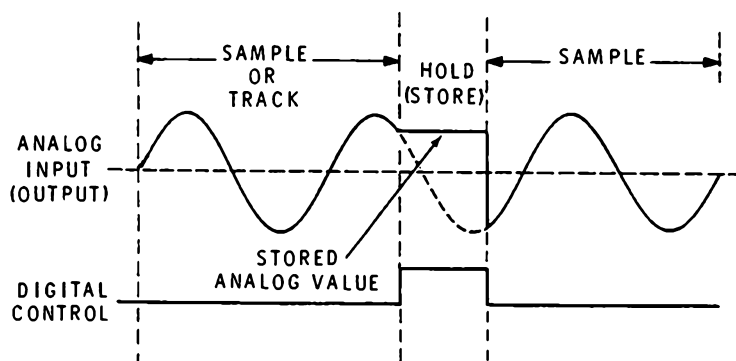


Figure 10-40

Input/output and control inputs of an S/H circuit.

Consider what happens when you apply a binary 1 to the switch. This turns the switch off and an open circuit now exists between the first follower and the capacitor. At this time, the voltage appearing across the capacitor will simply remain there. The very high input impedance of the second follower has little effect on the charge on the capacitor. Over a long period of time, the charge will leak off through the high input impedance of the follower. But for short periods of time (less than a second), the charge on the capacitor varies little. This voltage is a sample of the analog input at a specific period in time. The output of the second follower is a DC voltage representing the charge on the capacitor. As you can see, the capacitor and follower form a type of analog memory.

Applying a binary 0 to the switch will again turn it on and cause the analog signal to be applied to the capacitor. The circuit begins to sample or track the analog input.

This is a circuit that will allow an analog input signal to be passed to the output. However, by controlling the on/off state of the switch, the analog input signal can be sampled at any time. The value of that input is then stored temporarily.

Numerous other methods of S/H are used. However, the example given is sufficient to illustrate the concept. S/H circuits are widely used in all forms of data conversion. Their primary application is as an analog memory circuit connected to the input of an A-to-D converter as shown in Figure 10-39. Here, the sample and hold is used to freeze or sample the analog input and allow the A-to-D to do its conversion on a fixed voltage input. This removes the aperture error from the analog-to-digital conversion process. Note the special symbol used to represent the S/H circuit.

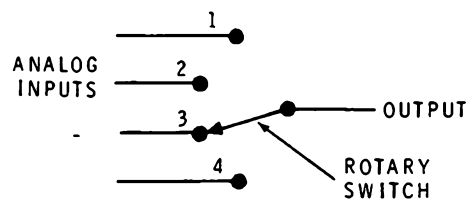


Figure 10-41

A mechanical analog multiplexer.

MULTIPLEXERS

A multiplexer is a multiple input, single output, digitally controlled analog switch. A mechanical analog multiplexer is illustrated in Figure 10-41. This is nothing more than a rotary switch where input signals are applied to the various positions of the switch. The arm of the switch may be connected to any of the inputs. Only one of the inputs is connected to the output arm at a time. An electronic version of a multiplexer is used to connect any one of several analog signals to the input of an ADC. See Figure 10-42. With this arrangement, a single ADC may be used to digitize any one of several analog inputs.

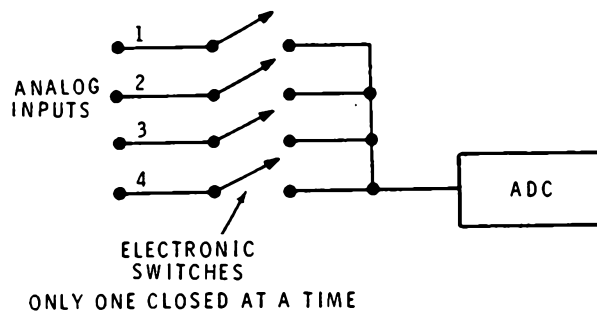


Figure 10-42

Electronic multiplexer.

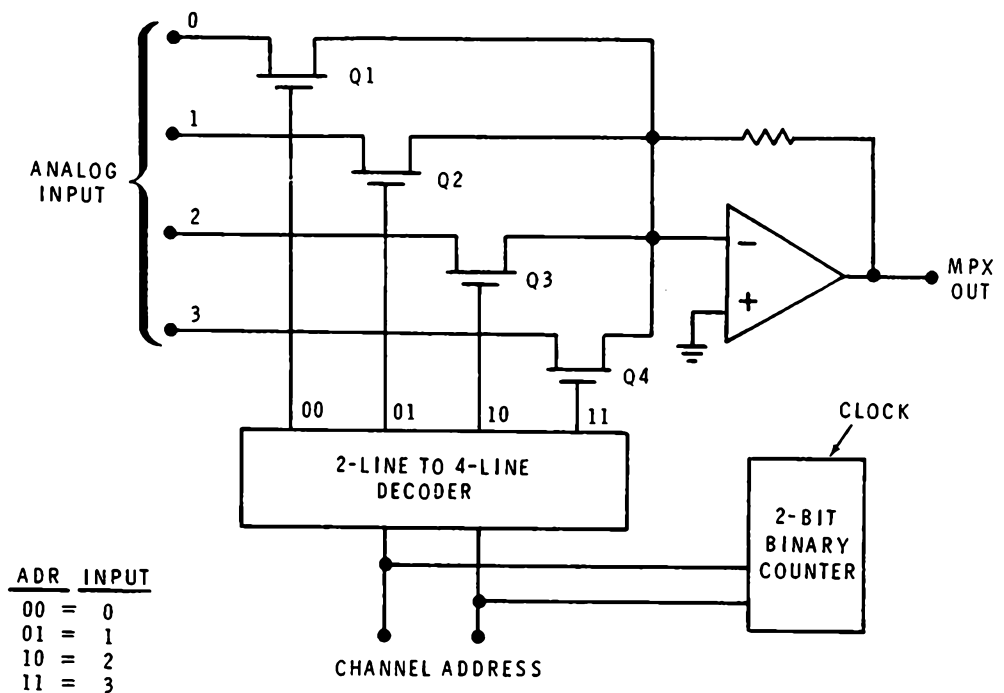


Figure 10-43

4-channel MOSFET multiplexer.

Analog multiplexers (usually abbreviated MPX or MUX) may be implemented in a variety of ways. Reed relays and bipolar transistors can be used as switches. However, most contemporary analog multiplexers are integrated circuits using field effect transistors. Both junction and MOSFETs are used. A typical 4-channel multiplexer using enhancement mode MOSFETs is shown in Figure 10-43. When the gate input to an enhancement MOSFET is at ground, the FET does not conduct. Therefore, it acts as an open circuit. When a binary 1 voltage level is applied to the gate, the FET conducts. Its low conducting resistance simulates a closed switch. In all multiplexers, only one of the FETs conducts at any given time. No more than one input signal is allowed to pass from the input to the output at any given time.

Channel selection is usually handled by an input register or binary counter and a decoder. In Figure 10-43, a 2-bit binary counter connected to a 2 line to 4 line decoder is used to select the desired input channel. Recall that with a decoder only one of the four outputs is enabled at any given time. There are four input states or addresses: 00, 01, 10, and 11. These correspond to the multiplexer input channels labelled 0 through 3. When input address 10 is applied, the decoder output turns on Q3 and lets analog input 2 pass through to the output. It is inverted by the op amp.

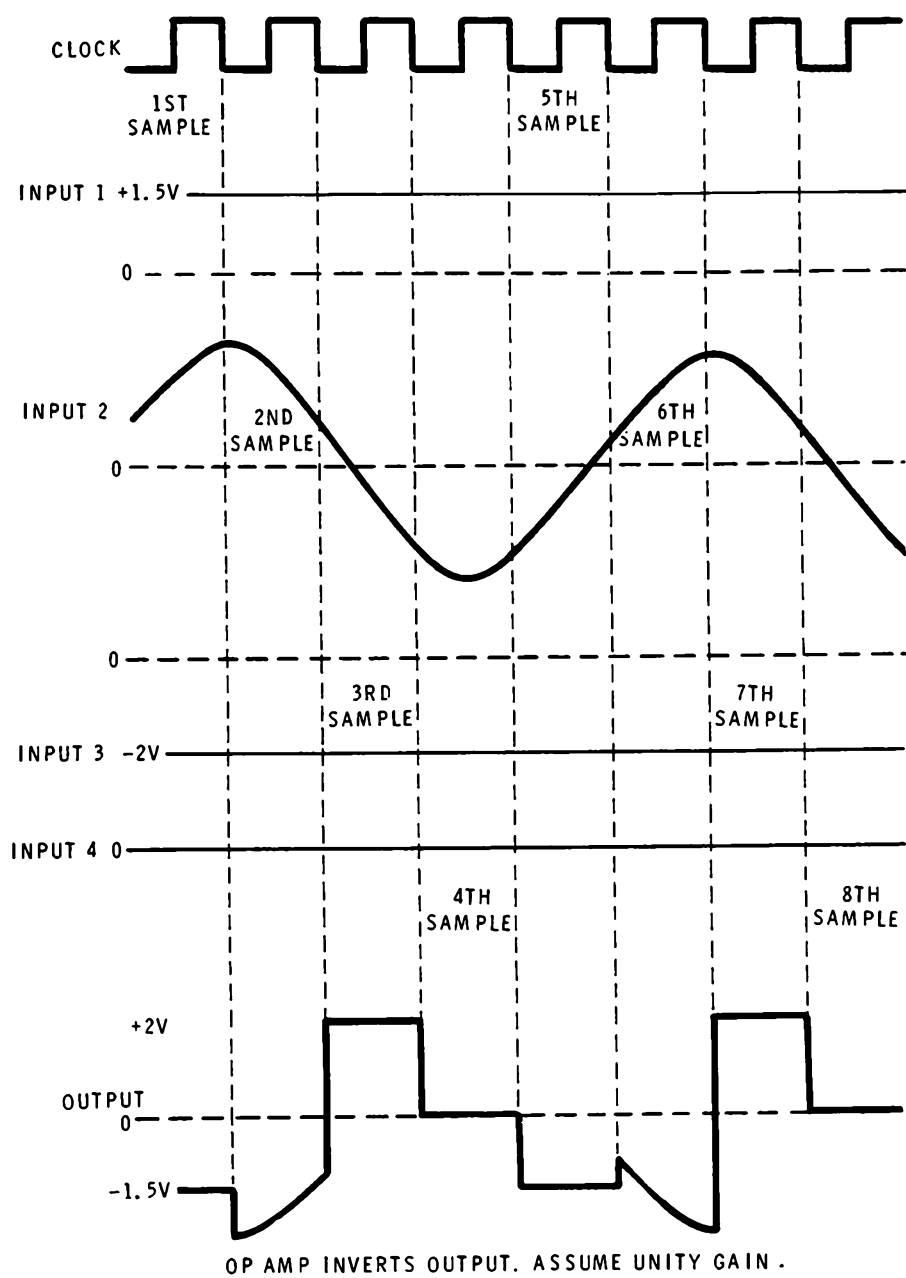


Figure 10-44

Output of a 4-channel mutliplexer
with given inputs.

With this circuit, incrementing the counter will cause each of the analog input channels to be sampled for a period of time for a duration equal to the period of the clock driving the counter. The waveforms in Figure 10-44 illustrate this concept. Notice that four different analog signals, +1.5 VDC, sine wave, -2 VDC, and zero volts are applied to the inputs. Note the multiplexer output is a repetitive sampling of the various inputs. The output is inverted because of the op amp. In a four-channel multiplexer, for every four clock pulses, a given input is sampled. If the clock frequency is 10 kHz, the sampling duration is $1/10,000$, or 100 microseconds, and the sampling rate for each input is $10 \text{ kHz}/4 = 2500 \text{ Hz}$.

Practical IC multiplexers are available with 2, 4, 8, and 16 input channels. These can be further combined to form multiplexers with even more inputs.

In some applications, it is desirable to sample multiple analog inputs simultaneously. Usually, the decoder is driven by a counter so that it samples each input channel sequentially, one after the other. In order to get readings of all four channels simultaneously, the analog inputs are applied to S/H circuits before they are applied to the multiplexer. This is illustrated in Figure 10-45. Here, the S/H amplifier outputs are applied to the multiplexer. At the desired time, all four S/H circuits are put into their hold mode with the S/H control input. This “freezes” the four inputs. Then the multiplexer can be cycled to sample the S/H circuit outputs. The ADC then sequentially converts the four inputs to their digital equivalents. While the analog-to-digital conversion process is sequential, the values obtained are the input levels of the four channels occurring at the desired instant in time. The process of sequentially sampling and converting multiple analog inputs is called time division multiplexing.

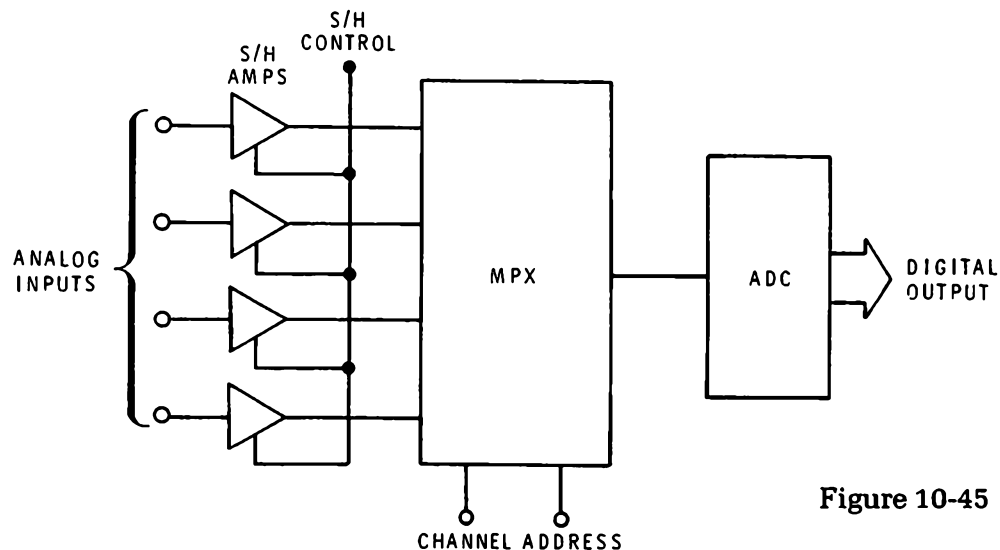


Figure 10-45

S/H used before a multiplexer.

Self Test Review

16. Most A-to-D converters contain a circuit that accepts analog inputs and generates a single digital output. It is called a _____.
17. Two types of ADCs that use a DAC are:
 - a. _____
 - b. _____
18. The three factors that affect the conversion speed of a counter feedback ADC are:
 - a. _____
 - b. _____
 - c. _____
19. A successive approximation converter with eight bits makes one comparison per microsecond. The conversion speed of this ADC is _____ microseconds.
20. The fastest form of ADC is the _____ converter.
21. The ADC with excellent accuracy and great resistance to noise is the _____ converter.
22. An ADC that converts a varying analog input into a train of pulses whose frequency is proportional to the input voltage is called a _____ converter.
23. The resolution of a 5-bit simultaneous converter with a 10 volt reference is _____ volts.
24. A circuit used to hold the voltage input to an ADC constant during its conversion cycle is called a _____.
25. A circuit that allows a single ADC to convert multiple analog inputs to digital outputs is called a _____.
26. The two types of ADCs commonly used in digital multimeters are:
 - a. _____
 - b. _____

27. The data conversion circuit that generates pulse code modulation is called a _____.
28. An ADC has an analog input signal of 400 Hz. The minimum sampling frequency is _____ Hz. The ideal sampling frequency is _____ kHz.
29. The inaccuracy caused by dividing the analog input range into discrete voltage levels is generally referred to as _____ error.
30. The inaccuracy produced into an analog-to-digital conversion by a varying analog input during the conversion process is referred to as _____ error.
31. An ADC that typically uses a BCD output is the _____ converter.
32. The maximum bit size of a modern flash converter in IC form is _____ bits.
33. An 8-input multiplexer is stepped by a 10 kHz signal. The individual sampling rate for each channel is _____ kHz.
34. The critical specification of a comparator is its _____ voltage.
35. An application for an ADC requires the circuit to recognize a minimum analog input of 12 millivolts. The reference is 10 volts. What size standard ADC could be used to satisfy this requirement? _____ bits
36. An A-to-D application requires a resolution of 16 bits. The conversion time must be less than 10 microseconds. A _____ converter is the preferred choice.

Answers

16. comparator
17. a. counter ramp
b. successive approximations

18.
 - a. analog signal amplitude
 - b. counter length (resolution)
 - c. clock speed
19. 8
20. flash
21. dual slope
22. V-F
23. 312.5 mV $5 \text{ bits} = 32 \text{ increments}$ $10 \text{ volt reference}$
 $10/32 = .3125 \text{ volts}$
24. S/H amplifier
25. multiplexer
26.
 - a. dual slope
 - b. V-F
27. CODEC
28. 800 Hz, 4 kHz
29. quantizing
30. aperture
31. dual slope
32. 8
33. $10 \text{ kHz}/8 = 1.25 \text{ kHz}$
34. offset or sensitivity
35. 10 bits, with 10 bits there are 1024 increments.
 $10/1024 = 9.8 \text{ mV resolution}$
With 9 bits, there are 512 increments.
 $9/512 = 19.5 \text{ mV resolution}$ which cannot recognize
12 mV changes.
36. successive approximations

EXPERIMENT 25

Analog-to-Digital Conversion

OBJECTIVES: *To demonstrate the operation of a typical analog-to-digital converter.*

Introduction

In the section covering analog-to-digital converters (ADCs), you read about the counter-ramp feedback ADC. Recall that this method takes the output of a DAC and compares it to a separate analog input. The output of the comparator controls the clock input to the counter which, in turn, provides the input to the DAC. The counter is incremented until the DAC output exceeds the analog input, at which time the comparator output prevents the counter from changing its input, and conversion stops.

This is the circuit you will work with in this experiment. Using the DAC circuit you wired in Experiment 24, you will add a comparator and appropriate control circuit to create an ADC. The list of required parts and the schematic in this experiment will show only those additional parts needed to convert the DAC to an ADC. Refer back to Figure 10-20A for the wiring of the DAC.

Material Required

- 1 — Wired DAC circuit from Experiment 24 (Figure 10-20A)
- 1 — ET-3200 Digital Design Experimenter
- 2 — 10k ohm resistors
- 1 — 4.7 volt zener diode
- 1 — 10k linear potentiometers
- 1 — 7400 quad two input gate
- 1 — multimeter (digital preferred)

Procedure

1. Before you add the additional circuitry to the DAC circuit already wired on your Trainer, check the circuit for proper operation. With the Trainer off, insert an ammeter between the 5 kilohm control and +5 volts. Turn your Trainer on and adjust the 5 kilohm control until the ammeter reads 2 milliamps. Once this adjustment is made, turn the Trainer off and replace the ammeter with jumper wire.

2. Move the clock input of the first counter to logic switch A, and ensure data switch 1 is low.
3. Turn the Trainer on and toggle the counter until L3 is lit, and the 7-segment LED displays an "8".
4. Connect a voltmeter to pin 1 of the 1458 op amp. Adjust the 10 kilohm control *on the feedback path of the op amp* until the output voltage equals +1.60 volts.

Once these adjustments are made, your DAC circuit should be operating precisely like the previous experiment.

5. Turn the Trainer off, and add the circuitry shown in Figure 10-46.

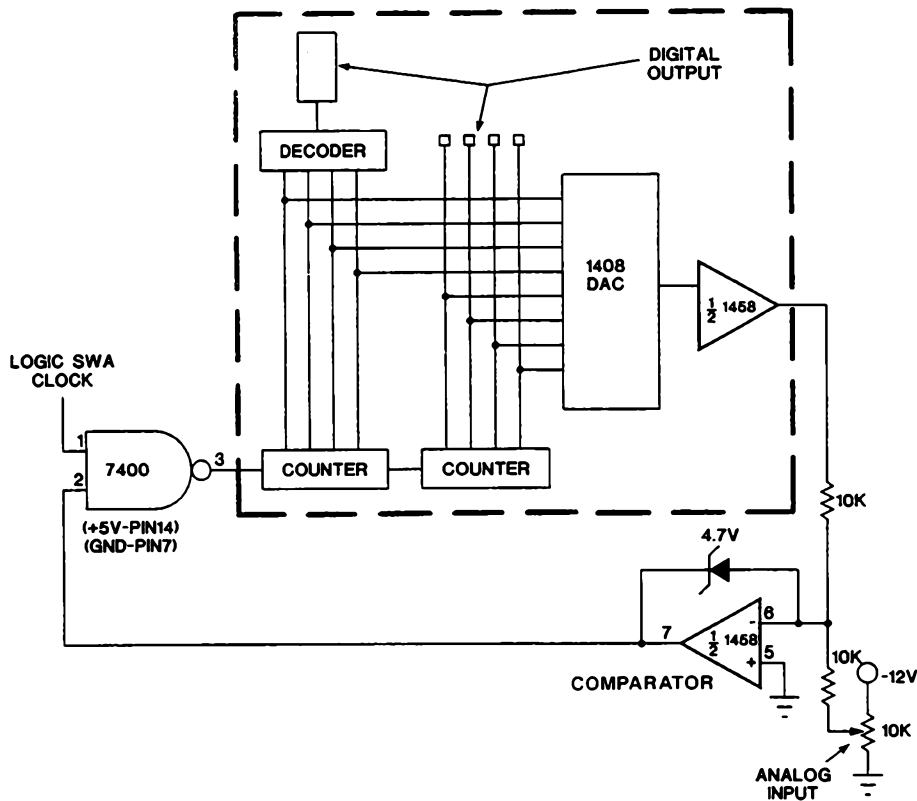


Figure 10-46

The comparator uses the second half of the 1458 dual op amp. One op amp (pins 1 through 3) acts as the current-to-voltage converter, while the other (pins 5 through 7) functions as the comparator.

Be sure to connect the cathode side of the 4.7-volt zener diode to pin 7 of the op amp.

In addition to the input and output pins of the 7400 chip, be certain to connect pin 14 to +5 volts, and pin 7 to ground.

Also be sure to remove the wire between the input of the first counter and logic switch A.

6. With the Trainer still off, set the analog input to -12 volts. This is accomplished by adjusting the 10 kilohm control *at the analog input* all the way to -12 volts. Connect the clock input at the NAND gate to CLOCK, and set the CLOCK frequency to 1 Hz. Set data SW1 high.

By adding a comparator, an analog input, and a clock control circuit (the NAND gate), the DAC circuit is converted to a counter-ramp feedback ADC. The positive output voltage of the DAC is compared to the negative analog input voltage by the comparator. As long as the voltage output of the DAC is lower in magnitude than the analog input, the anode of the zener is negative. This keeps the zener diode reverse-biased, and the output of the comparator will be V_{CC} , or a binary 1. This is one input to the NAND gate. As long as the output of the comparator is a binary 1, the NAND gate will allow the clock signal through, incrementing the counter circuits and changing the digital output.

Once the DAC output voltage becomes even nominally higher than the analog input, the zener becomes forward-biased. This drops the output of the comparator to +0.70 volts, or a binary 0. The 0 on the NAND gate locks the output of the gate to a binary 1, preventing the counter from cycling any further. In other words, once the DAC output exceeds the analog input, the counters cannot cycle and remain at their previous digital output.

7. Connect a voltmeter to the output of the DAC circuit, and turn your Trainer on. Move data SW1 low, allowing the counter to begin cycling. Measure and record the highest output voltage achieved by the DAC. This should occur when the binary input is 11111111.

(NOTE: Recall the method used in Experiment 24 to read the binary input. The four least significant bits is displayed by the hexadecimal display of the 7-segment LED. A binary 1111 is equivalent to an "F". The four most significant bits are displayed by logic indicators L4 through L1, with L1 being the MSB.)

Keep in mind that each voltage step lasts for approximately one second before the next clock pulse toggles the counters, so make your measurement quickly.

Output voltage (binary 11111111) = _____ V

8. Increase the frequency of the CLOCK generator from 1 Hz to 1 kHz. Are the counters able to go completely through their cycle?

(yes/no)

With the additional circuitry connected, the circuit is functioning as an ADC. There should have been a point where the output of the DAC exceeded the analog input, allowing the comparator to lock the counter. Yet the counters were never locked — they cycled through completely. How do you account for this?

9. Move data SW1 high, and set the frequency of the CLOCK generator to 1 Hz. Connect a voltmeter to the wiper arm of the 10 kilohm control that sets the analog input. Adjust the 10 kilohm control until the voltmeter reads -2 volts.
10. In Experiment 24, you determined the amount of voltage change for each incremental change in the binary input. Since you are using the same basic circuit here, this value must still be the same. Record this value below:

DAC Output = _____ V/step

11. Recalling the operation of this circuit as an ADC, the counters will start with a binary 00000000, and count until the comparator and NAND gate prevent the clock from reaching them. This should occur at a point where the output voltage of the DAC exceeded the analog input voltage. Since you know the amount of voltage change-per-step for the DAC, you should be able to predict the number of steps the counter will reach before locking up. This value would also be the binary input. Since we set the analog input to -2 volts, what will the binary output be?

Binary output (analog input = $-2V$) = _____

12. Set data SW1 low, and observe the operation of the circuit. Allow the counters to count until they stop. What is the binary output as displayed by the 7-segment LED and logic indicators L4 through L1?

Binary output = _____

Does the displayed binary output approximately equal the binary output you predicted in Step 11?

(yes/no)

13. Connect your voltmeter to the voltage output of the DAC. Does this voltage have the same magnitude as the analog input voltage?

(yes/no)

14. Set data SW1 high, and increase the frequency of the CLOCK generator to 1 kHz. Set data SW1 low, and note how long it takes the ADC to convert the analog input to its appropriate binary output. Was the conversion rate faster or slower than when the CLOCK generator was set to 1 Hz?

(faster/slower)

From this step, what can you conclude to be a factor that determines the conversion rate for an ADC?

Discussion

In Steps 1 through 14, you set up and began observing the operation of the counter-ramp feedback ADC. Since the counter-ramp ADC uses a DAC circuit, you were able to use the DAC circuit you constructed in Experiment 24. Initially, you ensured the DAC was working the same as in the previous experiment. You then attached the feedback loop consisting of the comparator and the NAND gate. Finally you added the required analog input signal, and the ADC was complete.

You started with an analog input of -12 volts. In Step 7, you measured the output of the DAC with a binary input of 11111111. This value should have produced an output of approximately $+10.36$ volts, just as in Experiment 24. You then increased the frequency of the CLOCK generator, which allowed you to see the counters cycling at a higher rate. Since the analog input was -12 volts, the output voltage of the DAC never reached this magnitude. Thus, the comparator never disabled the NAND gate output, and the clock pulses were always present at the counters. Remember, the counter-ramp ADC compares the DAC output voltage to the analog input voltage. When the DAC voltage exceeds the applied analog voltage, the comparator disables the NAND gate and prevents the clock pulses from reaching the counter. Since the two voltages were never even equal, this never occurred, and the counters were able to continually cycle.

In Step 9, you adjusted the analog input voltage to -2 volts. The comparator would then stop the counters when the DAC output voltage exceeded $+2$ volts. Since you knew the voltage-per-step for the DAC was $+0.04$ volts (from Experiment 24), you were able to predict that the DAC would exceed $+2$ volts in 51 steps, or when the binary count was 00110011. The actual binary output observed in Step 12 will depend upon how accurately you set the step voltage.

In theory, the counters should stop when the DAC output is equal to the analog input. However, we've continued to mention in this experiment that the counters stop when the DAC output *exceeds* the analog input — you verified this in Step 13. In actual operation, the DAC output must exceed the analog input in order for the comparator to actually switch its output.

The frequency of the clock pulses was 1 Hz, so the circuit took approximately 51 seconds to achieve an output that exceeded 2 volts. When you increased the frequency of the clock pulses in Step 14, the conversion took place almost instantaneously. This demonstrates that the frequency of the clock is one of the factors that determines the conversion rate for the counter-ramp ADC. In the next portion of this experiment, you will manipulate the circuit to observe the other two factors that determine the conversion rate for the counter-ramp ADC.

Procedure (continued)

15. Set data SW1 high, and connect a voltmeter to the wiper arm of the 10 kilohm control that sets the analog input. Adjust the 10 kilohm control until the voltmeter reads -6 volts. Set the frequency of the CLOCK generator to 1 Hz.

16. Based on the setting from Step 15, what should the binary output be?

Binary output (analog input = -6V) = _____

17. Set data SW1 low, and observe the operation of the circuit. Allow the counters to count until they stop. What is the binary output as displayed by the 7-segment LED and logic indicators L4 through L1?

Binary output = _____

Does the displayed binary output approximately equal the binary output you predicted in Step 16?

(yes/no)

18. Connect your voltmeter to the voltage output of the DAC. Does this voltage have the same magnitude as the analog input voltage?

(yes/no)

19. You have now seen the ADC convert analog inputs of -2 volts and -6 volts with the CLOCK frequency set at 1 Hz. Did the ADC convert the -6 volt input *faster* or *slower* than when the analog input was -2 volts?

(faster/slower)

Another factor, therefore, which determines the conversion rate for this circuit is:

20. Set data SW1 high, and connect a voltmeter to the wiper arm of the 10 kilohm control that sets the analog input. Adjust the 10 kilohm control until the voltmeter reads -1 volts. Set the frequency of the CLOCK generator to 1 kHz.
21. Move data SW1 low, and record the binary output displayed when the counters lock up:

Binary output = _____

22. Move data SW1 high, and adjust the 10 kilohm control *in the feedback path of the current-to-voltage converter* completely to one end.
23. Move data SW1 low, and record the binary output displayed when the counters lock up:

Binary output = _____

24. Move data SW1 high, and adjust the 10 kilohm control in the feedback path of the current-to-voltage converter to the other extreme end.
25. Move data SW1 low, and record the binary output displayed when the counters lock up:

Binary output = _____

26. Answer the following questions:

- a. Are the binary outputs the same for each adjustment of the feedback resistance?

(yes/no)

- b. As you adjust the feedback resistance, what characteristic of the DAC are you adjusting?

- c. How does this adjustment change the binary output of the ADC circuit?

- d. Given the answers to these and previous questions, the three factors which determine the conversion rate for this circuit are:

1. _____
2. _____
3. _____

27. Turn the Trainer and any test equipment off, and remove the circuit from the breadboard.

Discussion

Having worked with the normal operation of the counter-ramp ADC, Steps 15 through 26 demonstrated the other two factors which determine the conversion rate for the circuit. In Step 15, you adjusted the analog input to -6 volts. The binary output for this voltage is 10010111, which you viewed in Step 17. Again, the output voltage of the DAC in Step 18 should have been slightly higher than the analog input voltage.

In the first portion of the experiment, the ADC produced a binary output after adjusting the analog input to -2 volts. This conversion was accomplished with a clock frequency of 1 Hz. In Steps 16 and 17, you watched the conversion when the analog input was -6 volts and the clock was still 1 Hz. It took the circuit much longer to convert -6 volts to a binary signal than -2 volts. This is because the output voltage of the DAC increases slowly with each step, and the circuit requires more steps to reach $+6$ volts. So another factor that determines the conversion rate for the counter-ramp ADC is the magnitude of the analog voltage.

Steps 20 through 26 demonstrated the third factor for conversion rate. Working with a clock rate of 1 kHz, you adjusted the analog input to -1 volt. This input produced a binary output of 00011010 (decimal 26), allowing for the slightly higher output voltage of the DAC. You then recorded the binary output for the same analog input, but after adjusting the voltage-per-step of the DAC to its two extremes. Swinging the 10 kilohm control on the I-to-E converter changed the volts-per-step to anywhere from 0.08 to 0.01 . These two values produce binary outputs of 00001101 (decimal 13) and 10000011 (decimal 131). You measured the output voltage of the DAC for all three binary outputs, and noticed that it remained at relatively the same voltage. This output voltage was always slightly higher than the analog input voltage, but opposite in polarity.

So how did changing the volts-per-step change the binary output? Let's use an analogy to help explain. Imagine you wish to climb a flight of 12 steps. If you took one step at a time, you would take 12 steps before reaching the top. If you took two steps at a time, you would only need six steps to reach the top. If you took four steps at a time (and had very long legs), you could reach the top in only three steps.

Remember, the binary output is an indication of how many steps the DAC used before producing an output voltage slightly higher in magnitude than the analog input. When the volts-per-step is very high — 0.08 volts — the DAC will reach $+1$ volt very quickly. On the other hand, if the volts-per-step is only 0.01 volt, the DAC would require many more steps before reaching $+1$ volt. This proves that the volts-per-step of the DAC — also known as the **resolution** — is the third factor which determines the conversion rate of the counter-ramp ADC.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and work every problem first before checking the answers.

1. The increased use of data conversion circuits is the result of the increased application of:
 - a. op amps
 - b. digital equipment
 - c. computers
 - d. linear circuits
2. Which of the following does NOT affect the output amplitude of a DAC?
 - a. current switch speed
 - b. binary input
 - c. reference voltage
 - d. op amp feedback resistor
3. The primary disadvantage of a weighted resistor DAC is its:
 - a. low speed
 - b. wide resistor range
 - c. complexity
 - d. high cost
4. The resolution of a 9-bit DAC is:
 - a. .001953%
 - b. .1953%
 - c. 195.3 ppm
 - d. 1953 ppm
5. Which set of resistor values could be used in an R/2R DAC?
 - a. 10k, 15k
 - b. 300, 200
 - c. 600, 300
 - d. 600, 800

6. The term used to describe a DAC output that always increases for an increasing binary input count is:
 - a. accurate
 - b. linear
 - c. monotonic
 - d. monolithic
7. Which of the following is NOT a part of most DACs?
 - a. counter
 - b. reference
 - c. resistor network
 - d. current switches
8. The main function of an op amp in a DAC is:
 - a. current-to-voltage converter
 - b. analog memory
 - c. voltage-to-current converter
 - d. power amplification
9. The conversion speed of a DAC is essentially its:
 - a. aperture time
 - b. clock rate
 - c. resolution
 - d. settling time
10. A DAC that can produce the product of analog and digital inputs is called a(an):
 - a. product detector
 - b. multiplying DAC
 - c. hybrid DAC
 - d. A-D converter
11. The term used to describe the analog-to-digital conversion process is:
 - a. analogize
 - b. linearize
 - c. digitize
 - d. binaryize

12. The circuit with an analog input, reference input, and a binary output is called a:
- a. op amp
 - b. comparator
 - c. ADC
 - d. S/H
13. A counter-ramp ADC has an 8-bit resolution, a 10 volt reference, and a clock speed of 100 kHz. With a 6 volt input, the conversion time is:
- a. 10 microseconds
 - b. 154 microseconds
 - c. 1.54 ms
 - d. 2.56 ms
14. A 12-bit successive approximations ADC makes how many comparisons per conversion cycle?
- a. 1
 - b. 2
 - c. 12
 - d. 4096
15. The second fastest form of ADC is:
- a. successive approximations
 - b. counter ramp
 - c. flash
 - d. V-F
16. The fastest form of ADC is the:
- a. successive approximations
 - b. counter ramp
 - c. flash
 - d. dual slope
17. The form of ADC most often used in digital multimeters is:
- a. successive approximations
 - b. counter ramp
 - c. flash
 - d. dual slope

18. The primary circuit of a flash converter is a:
- a. comparator
 - b. DAC
 - c. op amp
 - d. R/2R network
19. The primary advantages of a dual slope ADC are:
- a. conversion speed
 - b. accuracy
 - c. high noise tolerance
 - d. low cost
20. The circuit that converts the output of a V-F ADC to binary or BCD is a:
- a. counter
 - b. shift register
 - c. storage register
 - d. DAC
21. The error caused by the analog input changing during an ADC cycle is called:
- a. monotonic error
 - b. quantizing
 - c. reference error
 - d. aperture error
22. An analog memory circuit used to eliminate aperture error is called a:
- a. multiplexer
 - b. track/store amplifier
 - c. comparator
 - d. op amp

23. An 8-channel multiplexer is driven by a 20 kHz clock. The optimum sampling duration is:

- a. 6.25 microseconds
- b. 5 microseconds
- c. 100 microseconds
- d. 400 microseconds

and the minimum sampling speed is:

- a. 2.5 microseconds
- b. 25 microseconds
- c. 250 microseconds
- d. 20 microseconds

24. An LSI IC that generates PCM is called a:

- a. successive approximations ADC
- b. modulator
- c. CODEC
- d. modem

25. The process of sequentially converting multiple analog input signals to digital outputs is called:

- a. analog-to-digital conversion
- b. serial ADC
- c. pulse code modulation
- d. time division multiplexing

EXAMINATION ANSWERS

1. c.—computers
2. a.—current switch speed
3. b.—wide resistor range
4. b.—.1953%, d.—1953 ppm
9 bits = 2 to the 9th power = 512
 $.001953 \times 100 = .1953\%$
 $.001953 \times 1,000,000 = 1953 \text{ ppm}$
5. c—600, 300
6. c—monotonic
7. a.—counter
8. a.—current-to-voltage converter
9. d.—settling time
10. b.—multiplying DAC
11. c.—digitize
12. b.—comparator
13. c—1.54 ms
Resolution: 2 to the 8th power = 256
 $1/256 = .0039$ With 10 volt reference, resolution is
 .039 V or 39 mV.
With a 6 volt input, the counter must increment
 $6/.039 = 154 \text{ steps.}$
With 100 kHz clock, the step speed is $1/100,000 =$
 .00001 or 10 microseconds
Conversion speed is $10 \times 154 = 1540 \text{ microseconds}$
 or 1.54 milliseconds (ms).

14. c.—12

15. a.—successive approximations

16. c.—flash

17. d.—dual slope

18. a.—comparator

19. b.—accuracy, c.—high noise tolerance

20. a.—counter

21. d.—aperture error

22. b.—track/store (or S/H) amplifier

23. b.—5 microseconds $T = \frac{1}{FX10}$ (optimum)

$$1/200,000 = .000005 \text{ sec or 5 microseconds}$$

$$T = \frac{1}{FX2} = \frac{1}{20kX2} = \frac{1}{40,000}$$

$$T = 25 \text{ microseconds}$$

24. c.—CODEC

25. d.—time division multiplexing

Unit 11

**DIGITAL
TROUBLESHOOTING**

CONTENTS

Introduction	11-3
Unit Objectives	11-4
Unit Activity Guide	11-5
Typical Problems in Digital Circuits	11-7
Digital IC Problems	11-19
Digital Test Instruments	11-25
Procedures for Digital Troubleshooting	11-63
Experiment 26 — Practical Digital Troubleshooting	11-81
Unit Examination	11-91
Examination Answers	11-99

INTRODUCTION

In this unit, you will learn to troubleshoot digital circuits. Like other circuits, digital circuits occasionally fail, and often, the problem must be located quickly so that the equipment can be restored to normal operation. The whole purpose of this unit is to help you learn how to diagnose, isolate, and repair defective digital equipment.

Whether you are an engineer, a technician, or a scientist, you should know how to troubleshoot digital circuits. As an engineer, you will frequently have to troubleshoot and repair defective digital circuits in your design prototypes. As a production line technician, you will occasionally need to repair new digital equipment that has just been assembled on the production line. As a scientist, you may have to service the piece of digital equipment you are using in your experiments. These are only a few of the many examples of situations where a knowledge of digital troubleshooting is desirable. In any case, the main objective is to locate the problem and repair it quickly to restore it to proper operation. Speed is necessary since equipment malfunctions cause a significant loss of time, money, and productivity. In this unit, you will learn troubleshooting practices and test equipment applications that will help you troubleshoot and repair digital circuits in newly constructed equipment or in equipment that has been used in the field.

UNIT OBJECTIVES

When you complete this Unit, you will be able to:

1. Name the common troubles that cause digital equipment to malfunction.
2. Explain how and why digital circuits fail.
3. List the special pieces of test equipment used to troubleshoot digital circuits.
4. Show how to troubleshoot digital circuits using common test instruments such as the VOM/DMM and the oscilloscope.
5. Explain the operation and application of logic probes, logic pulsers, logic analyzers, and signature analyzers.
6. List several common procedures for diagnosing and isolating the circuit problems.

UNIT ACTIVITY GUIDE

	Completion Time
<input type="checkbox"/> Read "Typical Problems in Digital Circuits."	_____
<input type="checkbox"/> Answer Self Test Review questions 1 – 10.	_____
<input type="checkbox"/> Read "Digital IC Problems."	_____
<input type="checkbox"/> Answer Self Test Review questions 11 – 15.	_____
<input type="checkbox"/> Read "Digital Test Instruments."	_____
<input type="checkbox"/> Answer Self Test Review questions 16 – 40.	_____
<input type="checkbox"/> Read "Procedures for Digital Troubleshooting."	_____
<input type="checkbox"/> Answer Self Test Review questions 41 – 50.	_____
<input type="checkbox"/> Perform Experiment 26 "Practical Digital Troubleshooting."	_____
<input type="checkbox"/> Complete the Unit Examination.	_____
<input type="checkbox"/> Check the Examination Answers.	_____

TYPICAL PROBLEMS IN DIGITAL CIRCUITS

Figure 11-1 lists the ten problems that most often cause trouble in digital equipment. While problems can come from a wide variety of sources, you will find that most of them fall into one of these ten categories. In this section, we will consider each of these classes of trouble in more detail.

1. Operator problems.
2. Construction errors.
3. Defective components.
4. Mechanical problems.
5. Power supplies.
6. Timing problems.
7. Environmental problems.
8. Noise.
9. Design errors.
10. Software.

Figure 11-1
Representative problems in digital equipment.

Operator Problems

Operator problems, more commonly referred to as “cockpit” problems, are one of the most frequently encountered troubles. Cockpit problems usually occur when the operator of the equipment uses it improperly. When the operator of the equipment doesn’t know how to use the equipment or interpret the results of the application, he often suspects the equipment to be defective when it isn’t. What appears to be a malfunction is, in reality, nothing more than the improper application of the equipment. Cockpit problems are not equipment problems, they are operator problems.

Some typical cockpit problems include:

1. **Improper interconnections** — The user has connected the equipment to external devices in an incorrect manner. For example, inputs may be connected to outputs and vice versa.
2. **Improper control settings** — The switches and other controls on the equipment have been misadjusted or set to incorrect positions.
3. **Output results are misinterpreted** — Output displays such as indicator lights, numerical readouts, and meters are read incorrectly.
4. **Incorrect data is used** — In equipment requiring the operator to enter instructions or data, input mistakes occasionally cause a malfunction.
5. **Equipment specifications exceeded** — The equipment is used in such a way that its specifications and operating characteristics are exceeded in some way.


Cockpit problems are relatively easy to overcome. It is primarily a matter of the operator reading the equipment instruction manuals. In addition, someone who knows how to use the equipment properly, should demonstrate for the operator. Once the user becomes familiar with the operation and limitations of the equipment, cockpit problems typically go away.



Construction Errors


Many problems occur in digital circuits that have just been constructed. When you are building a design prototype or manufacturing a new piece of equipment on the production line, construction mistakes are often introduced. Of course, construction errors occur only in newly manufactured items, as opposed to equipment that has been operating in the field for some time.

A typical example of construction problems is wiring errors in a design prototype. Usually when a circuit is designed, it is breadboarded and tested to ensure its proper operation and determine its specifications. Problems are introduced by the person constructing the circuit. Some typical problems include using the wrong integrated circuit, using an incorrect component value, and incorrectly connecting the components.



Construction errors are also found in newly manufactured equipment. While there are fewer mistakes made on the production line than on design prototypes, still you will find incorrect or defective components, wiring mistakes, and problems resulting from poor construction techniques. Examples of poor construction techniques are bad soldering and improper component placement. When components are soldered by hand, occasionally some connections will be missed entirely and, because of the lack of solder, an open circuit exists. Cold solder joints and solder bridges between adjacent connections are also common. Also, the builder may use incorrect components, or connect them incorrectly. For example, he may put an integrated circuit, a diode, or electrolytic capacitor in backwards.

While many construction errors are easy to find and correct, sometimes they can be extremely difficult. This is particularly true when there are multiple mistakes. It is difficult enough to locate a single problem, but it is much harder to isolate two or more independent defects. It is difficult to eliminate all construction errors but obviously the way to minimize them is to take the time and effort necessary to ensure that the equipment is constructed properly in the first place.



Defective Components

The most common cause of failure in digital equipment, or for that matter any electronic circuit, is a defective component. This is particularly true in equipment that has been operating successfully.

Electronic components fail for a number of reasons; most often they have been improperly used. For example, the component may be overloaded or its specifications exceeded due to poor design. Another reason for component failure is a manufacturing defect. The manufacturer may have simply put the component together incorrectly or used the wrong materials.

Some types of components fail more often than others. Component failure rates vary widely depending upon the type. Figure 11-2 lists the common components used in digital equipment. These are listed in the order of the most frequent failures. For example, fuses and indicator lights fail more frequently than other components. The component that fails the least is the printed circuit board. Review the list in Figure 11-2 to familiarize yourself with the most common component failures.

1. Fuses, indicator lights.
2. Switches, relays.
3. Power supplies.
4. Connectors, wires, cables.
5. Transistors, diodes.
6. Capacitors, resistors, transformers
7. Integrated circuits.
8. Printed circuit boards.

Figure 11-2
Component failure table.



Mechanical Problems

According to Figure 11-2, mechanical or electromechanical components fail more frequently than most other components. Devices such as switches, relays, and connectors are mechanical in nature and have a higher failure rate than electronic components. Any component that has moving parts or is used as a physical link between pieces of equipment causes more trouble than an electronic component. A good example of a mechanical problem is a broken wire. Loose and dirty interconnections are also common problems. Plugs, sockets, and jacks that are used with wire to connect one piece of equipment to another are frequent causes of difficulty. Often, these components fail to make a good electrical connection because they become worn or their contacts become corroded or dirty.


Any component with a moving part, such as a switch or relay, is a candidate for problems. Moving parts wear, get out of adjustment, and break because of stress. You will find yourself making more mechanical repairs than almost any other type.



Power Supplies

While most problems in electronic equipment can be traced to a component failure of some kind, very frequently that component is part of a section common to all electronic equipment, whether it is analog or digital. That circuit is the power supply. This is the circuit that converts the AC line voltage to accurate and stable DC voltages for operating the circuits in the equipment. When the power supply fails, the equipment becomes inoperable. Since it affects all of the circuits, a power supply problem is usually easy to isolate and repair.

Power supplies often fail because of the great stress put on them. They normally have to handle high voltages, high currents, or both. High temperature is another typical cause of failure. So you will probably spend a lot of your time repairing power supplies. A quick check of the circuit voltages will tell you whether or not the power supply is working. If the voltages are incorrect, you will typically know that you have a power supply problem.



Timing Problems

Timing problems are those associated with the frequency of operation and propagation delay of digital circuits. Most digital circuits operate at very high speeds, and the sequence of events and their speed of occurrence are usually essential to proper operation. If the clock frequency changes or a component develops increased propagation delay, operational problems can occur. Sometimes component aging or mechanical vibrations can introduce voltage or frequency changes and affect the timing. In any case, you will occasionally encounter timing problems in your work with digital circuits.

Most timing problems typically occur in circuit prototypes. When an engineer is designing a new circuit, he usually discovers malfunctions that are due to improper timing. Circuits with critical timing are usually redesigned or carefully designed to ensure that they function properly with a wide range of components and operating conditions. In this way, the unit can be successfully mass produced using off-the-shelf components with their wide range of characteristics and tolerances.


Timing problems are sometimes difficult to isolate. What may appear to work in a logic diagram may not, in reality, function properly because of component frequency limitations and wide variations in propagation delays. Even though digital components are fast, they do not have infinite speed and are not perfect. Most timing problems will show up in the design stages and can be corrected once they are isolated. Timing problems rarely occur in equipment that has been in the field for some time. Many times when timing problems occur, you can correct them by replacing a component or by making a frequency or time delay adjustment.



Environmental Problems


Environmental problems are those caused by the conditions of the environment in which the equipment is operating. A dirty environment is a good example. Some digital equipment is used in industrial environments where oil, grease, dust, chemicals, and salt air can affect the way equipment operates. Typically a dirty environment will introduce more mechanical failures than usual. Dirty and corroded connector pins and switch contacts are typical of the items that fail.

An environment in which considerable physical stress takes place can also cause troubles. An example of high physical stress is equipment vibration. Excessive vibration can cause mechanical failures of all types.



One of the more common environmental problems is temperature. Most electronic equipment is designed to operate properly over a relatively narrow temperature range. This range is typically from 0 to 70 degrees C. If the temperature is excessively low or high, the equipment may not operate properly. Virtually all electronic components are sensitive to extremes of temperature. Solid-state devices such as diodes, transistors, and integrated circuits will fail completely at extremely high temperatures, and many of these same electronic components will not work properly if the temperature is too low.

Excessive heat, rather than cold, is the more common problem. Most equipment is designed to help minimize or eliminate heat generated by the equipment. Heat sinks are used on high-powered transistors and integrated circuits to help dissipate the heat. Ventilating holes and fans are often used to circulate the air to keep the heat from building up. But despite the action taken by the designer to minimize heat buildup, most digital equipment will run hot.



Noise

Noise is the name given any random or unwanted electrical signal in the equipment. Noise can be introduced by an external source or may be generated internally. In any case, noise represents unwanted signals that cause a variety of troubles in digital equipment.

Some of the more common sources of noise are voltage spikes on the AC power line, current and voltage surges in the power line or power supply, magnetic or electrostatic fields or RF interference that comes from radio or TV transmitters. Noise is also generated by the high-speed switching action of logic circuits. All of these sources create noise that can cause the malfunction or total breakdown of the digital equipment.

Often the source of noise is the equipment itself. Internally generated noise can come from the power supply or from capacitive or inductive coupling of adjacent circuits. Improper or inadequate power supply filtering or regulation, or noise spikes generated by circuits being too close together are common causes. Noise problems are one of the most difficult to isolate. Noise is usually random and sometimes causes intermittent operation. As a result, noise is often difficult to track down. Once the source is located, the repair is generally easy. Most noise problems can be avoided if the equipment designer takes the potential noise problems into consideration during the design stages.

Design Errors

In new equipment and prototypes, design errors are sometimes a cause of trouble. Circuits that were designed to accomplish some specific task may not operate correctly due to mistakes made by the designer. Engineers occasionally miss important details or fail to anticipate unusual conditions and, therefore, make mistakes in their designs.

Luckily, most design errors are discovered during the prototype stage or as final tests and evaluations are being made. But there are times when design mistakes slip through and eventually show up in production units. This is true of very complex systems and sophisticated equipment where there are many modes of operation, critical specifications or severe operational conditions. Design errors must be found if equipment reliability and functionality is to be achieved. Only exhaustive testing and practical use can find design errors.

Software

In digital equipment using computers or in systems that can be programmed, software rather than hardware can be the cause of a problem. Software is a term used to refer to the programs that a computer executes. Programs are lists of binary coded instructions, usually stored in RAM or ROM, that tell a computer or other digital hardware what to do. If there is a software error where a wrong instruction is used or the instruction sequence is incorrect, the equipment will not function properly. The hardware may be operating correctly, but the software defect makes it appear as though a hardware problem is the cause. The troubleshooting challenge in programmed digital equipment is to determine if the problem is hardware or software.

Self Test Review

1. In digital equipment, _____ components fail more often than electronic components.
2. Operator misuse is called a _____ problem.
3. In newly constructed equipment, _____ errors are often the problem.
4. Problems with propagation delay and clock frequency are called _____ problems.
5. Three common sources of noise are:
 - A. _____
 - B. _____
 - C. _____
6. To minimize equipment "down time", _____ of repair is important in digital troubleshooting.
7. Integrated circuits fail less often than transistors.
 - A. True
 - B. False
8. The most common environmental problem is excessive _____.
9. In computers, a _____ problem can give the impression of a hardware failure.

10. List the following components in the order (1, 2, 3, etc) you would check them if you were troubleshooting a piece of digital equipment. Hint: List them in sequence with most likely failures first.

- PC board
- Switch
- PC board connector
- Power supply

Answers

1. mechanical
2. cockpit
3. wiring
4. timing
5.
 - A. Radio/TV transmitters
 - B. Power line spikes
 - C. Internal logic switching
6. speed
7.
 - A. True
8. heat
9. software (or programming)
10.
 1. Switch
 2. Power supply
 3. PC board connector
 4. PC board

DIGITAL IC PROBLEMS

While equipment failures come from a variety of sources, as you have seen, the problem you are most likely to encounter is a defective digital integrated circuit. There are more integrated circuits in digital equipment than any other type of component. In fact, you can almost count on better than 90% of all components being digital ICs of some sort. Because of this, when a problem occurs, it is most likely a digital IC. A high percentage of digital troubles in prototypes and newly manufactured equipment are caused by defective ICs. In this section, we discuss the ways digital ICs fail. The two types of IC failures you will encounter are classified as either external or internal. External problems are those associated with the physical installation and interconnection of the integrated circuit. Internal failures are defects that occur inside the IC. You will encounter both types in your troubleshooting.

Digital IC failures are generally easy to deal with. External failures can normally be very quickly and easily corrected. Internal failures, of course, cannot be. Once you have determined an IC has failed internally, all you can do is replace it. The challenge is in finding the defective IC. By knowing the ways ICs fail, you will be better prepared to find and deal with these problems.

External IC Problems

External IC failures often cause the integrated circuit itself to appear defective. The problems and symptoms of external and internal problems are usually very similar. The challenge in troubleshooting is to determine just where the problem is located.

Some typical examples of external IC problems include short circuits between IC pins, open or broken circuits, physical damage, or a problem with another component connected to the IC.

The most common external problem is a short between pins. The pins on a dual in-line integrated circuit package are spaced very closely. Occasionally when the integrated circuit is installed in a socket or printed circuit board, the adjacent pins can touch, but the most usual cause of a short between pins is a solder bridge. When adjacent pins are soldered, solder may accidentally flow between the two pins. A tiny hairline bit of solder can also short adjacent pins. Such hairline shorts are not often visible and are sometimes difficult to isolate.

Most digital ICs are basically insensitive to such connections. After the short has been corrected, the IC will usually work normally. However, there are shorts between pins that can cause damage. These are cases where circuit outputs could be shorted to either the supply voltage or ground. Here, an external problem damages the IC internally.

Another cause of shorts between adjacent pins is stray debris. Occasionally a tiny piece of wire, metal, or other scrap will stick to the bottom of a PC board, or lay on top of it, causing a short between pins. Shorts can also be caused by chemicals. Occasionally chemicals are used near production equipment and can be accidentally deposited on the printed circuit board. Such chemicals can create a short or a low resistance path between the pins.

A high salt environment can also cause low resistance paths between conductors. Salty air will cause a salt buildup on sockets, printed circuits or other metallic conductors. The buildup is usually gradual, and sometime in the future, a low resistance or short will occur.

Another common external failure is an open circuit. This could be caused by an improperly soldered pin. When an IC is installed on a printed circuit board, solder may accidentally have been left off the pin. A poor or "cold" solder joint will cause the same effect as an open circuit.

Open circuits can also be caused by a defective IC socket. In many designs, the integrated circuits are plugged into sockets. This makes them easy to remove and replace, which greatly aids in troubleshooting and repair. But when IC sockets are used, there is always the possibility that one of the pins of the socket may be defective. The result is usually an open circuit where the IC pin does not make proper contact with the socket pin soldered to the PC board.

Open circuits may also occur because of a defective PC board. The copper on the PC board may be open due to a manufacturing defect or a deep scratch. If wires are used to interconnect ICs or their sockets, a broken wire or poorly soldered wire can also cause an open circuit.

Physical failure or improper IC replacement can also cause trouble as well. For example, when a piece of equipment is manufactured, often the integrated circuit is installed backwards on the PC board. This is also true when an integrated circuit is installed in its socket. The IC will fit on the PC board or in the socket two ways. Of course, only one way is correct.

You can often correct the problem quickly by unplugging the IC from the socket and turning the IC around. This is more difficult to do when it is soldered to a PC board. In either case, the IC is rarely damaged if this is done. Turning the IC around solves the problem.

Sometimes when an IC is plugged into a socket, the IC pins do not enter the holes properly. The IC pins may be bent or broken, thus creating an open and sometimes a short circuit. Occasionally, an IC pin will miss the hole and will be forced to the outside of the socket, not making the connection.

Another type of external IC problem is the failure of a component connected to the IC. In many cases, the integrated circuit is connected to a switch, an indicator light, a connector, or other electronic components such as transistors, diodes, capacitors, and the like. All of these components can fail, creating either an open circuit or a short circuit. Often the problem will appear to be a defective IC, when in reality, it is the component connected to the IC that has failed.

Some of the more common problems include open or shorted capacitors, open or shorted diodes or transistors, or defective switches and indicator lights. Connectors are one of the more common sources of external IC problems. We have already mentioned IC sockets, but keep in mind that circuit board connectors and cable connectors also cause their share of problems.

Internal IC Problems

Internal IC problems fall into three basic categories: direct component or circuit failure, short circuits, and open circuits. In all cases, the component is defective and must be replaced in order to restore operation. It is simply impossible to repair a digital IC.

One of the most common causes of IC failure is simply a defective circuit. The circuit on the silicon chip itself is defective and results in a completely inoperative IC. Such failures can be caused by improper use of the component. For example, excessive voltages or currents can be applied to the IC, causing a component to fail. Extreme environmental conditions such as excessive heat or a short of the output to the supply voltage or ground can permanently damage an IC.

Alternatively, the chip itself may never have functioned correctly. Because of a flaw in the manufacturing process, the IC may have been defective when it was delivered. Most ICs are 100% tested before leaving the factory, but some defective ones still slip through.

Short circuits can cause internal defects. Just as shorts occur externally, they can also occur internally for most of the same reasons. In most ICs, the tiny silicon chip is attached to the IC pins by very fine wires. These fine wires are welded to the chip and to the pins. Occasionally, one of the welded bonds may come free, causing the tiny wire to touch an adjacent wire. A short is thus created. Stray debris may also cause such a problem, although it is not common. In addition, shorts can occur on the chip itself. Because of high currents or voltages, the chip can become damaged and adjacent connections can become shorted together.

Open circuits are also common in internal failures. A bond may come unwelded at either the pin or the chip, thus creating an open circuit. Open circuits can also occur on the chip itself. Overloads of current or voltage can cause the fine wires connecting the chip to the pins to burn through. Open pins may be the most common internal IC problem.

Figure 11-3 shows several typical internal IC problems.

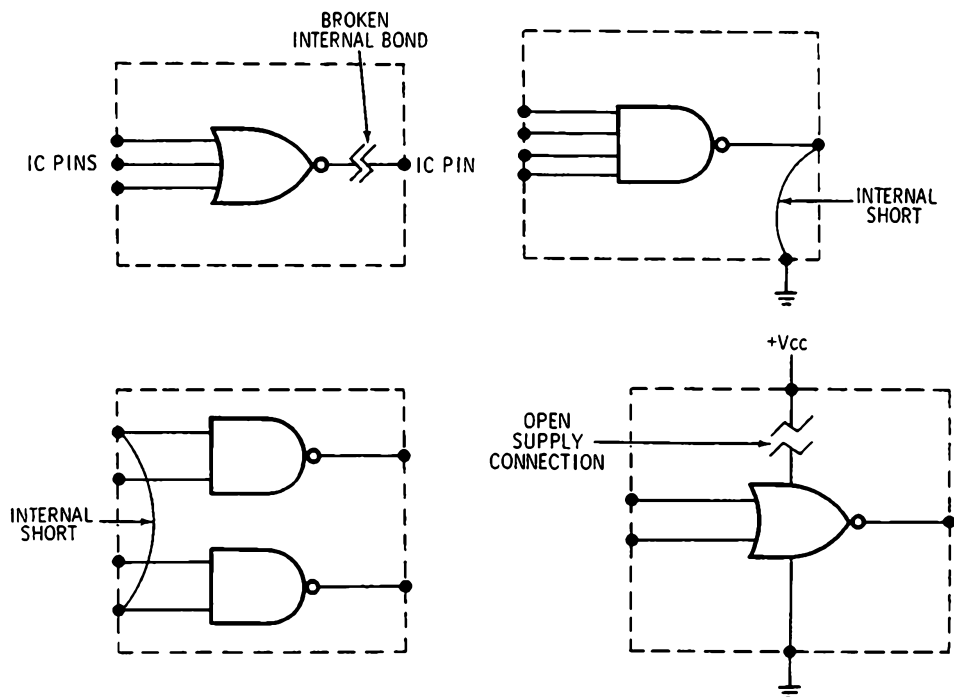




Figure 11-3
Representative internal IC problems.



Self Test Review

11. The two types of IC failures are _____ and _____.
 12. Internal IC failures can only be overcome by _____.
 13. List three common external IC problems:
 - A. _____
 - B. _____
 - C. _____
 14. The three major internal IC problems are:
 - A. _____
 - B. _____
 - C. _____
 15. The most common cause of an external short is a _____
_____.
- 
- 

Answers

- 11. internal, external
- 12. replacement
- 13. A. Poor soldering.
B. Bent or broken pins.
C. IC in backwards.
- 14. A. Circuit failure.
B. Opens.
C. Shorts.
- 15. solder bridge

DIGITAL TEST INSTRUMENTS

Most standard test equipment such as multimeters and oscilloscopes can be used in troubleshooting digital equipment. However, there are also special test instruments designed specifically to make it faster and easier for you to locate digital logic troubles. In this section, we describe the specification and application of standard test instruments for digital troubleshooting. In addition, we introduce a variety of special digital test instruments such as logic probes, logic pulsers, logic analyzers, and signature analyzers. In a later section, you will see how these test instruments can be used in locating a variety of digital problems.

Multimeters

Standard analog or digital multimeters like those listed in Figure 11-4 are used in troubleshooting digital equipment. The most often used application is for checking AC line voltage and DC power supply voltages in digital equipment. You will also use the resistance measuring capability of a multimeter for continuity checking. For example, you can test for open and shorts in wires and cables. You can also check connector pins, test fuses, and verify that incandescent lamps are open or good. The current measuring capability of a multimeter is rarely if ever used in testing and troubleshooting digital circuits.

IM-2262 or 2264 or IM-2215	Digital
IM-5217	Analog

Figure 11-4
Standard analog and digital multimeters are useful in digital troubleshooting.

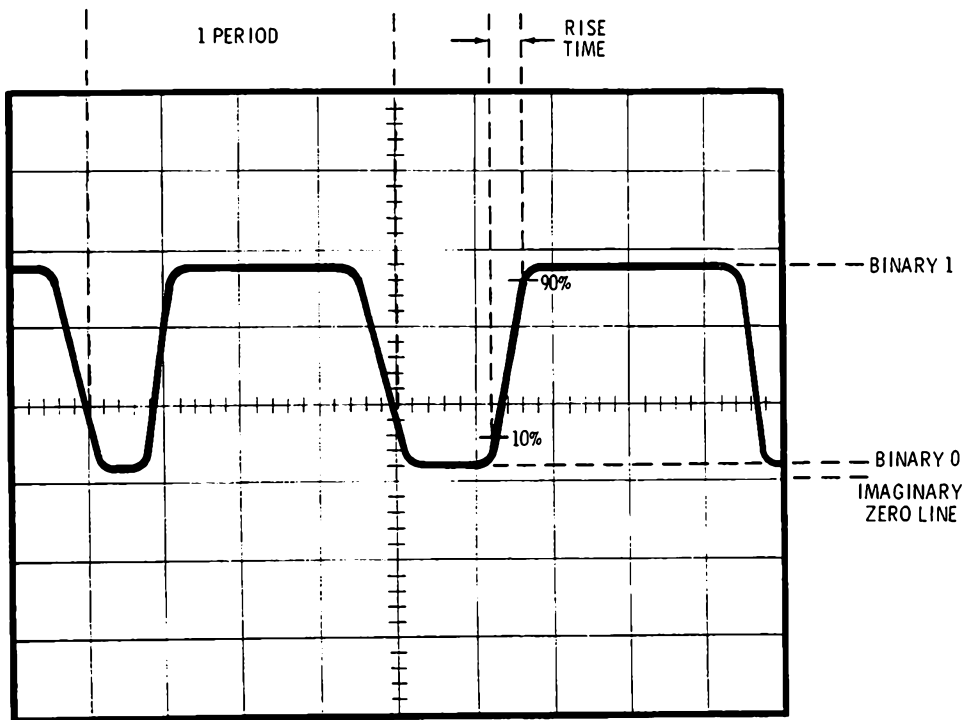
You can also use the multimeter to monitor logic levels in digital circuits. You can use almost any multimeter to check for the existence of binary 0's and binary 1's. Analog meters are best for just looking for 1's and 0's, as the meter pointer is easy to see and follow. Digital meters are best when you want or need to know specific values. Multimeters with good low voltage resolution are the most useful. Most digital logic levels are relatively low voltages, 5 volts or less, and a meter capable of measuring such voltages accurately is of benefit. Typical binary 0 levels are often only several tenths of a volt. A meter with good low voltage resolution will give the best results. For very accurate measurements, a digital multimeter is preferred.

Oscilloscopes

Multimeters are used primarily in static testing. That is, when a multimeter is being used it is common to check for fixed or static logic levels. When digital equipment is operating at high speed, a multimeter is basically useless in monitoring logic levels. An oscilloscope is a dynamic test instrument that allows you to see what is going on when the equipment is operating at high speeds. For that reason, oscilloscopes are widely used in troubleshooting digital equipment.

Since most oscilloscopes are calibrated for both amplitude and time measurements, you can measure almost all characteristics of digital signals. For example, you can measure logic levels quickly on an oscilloscope. Vertical calibrations on the graticule represent fixed amounts of voltage and, therefore, provide a quick and accurate indication of both binary 0 and binary 1 levels. The horizontal graticule calibration represents units of time. For this reason you can use the scope to measure rise and fall times, propagation delay, pulse width, and duty cycle.

Since time measurement can be translated into frequency ($f = 1/t$), you can also measure frequency with an oscilloscope. A common application is to check the clock signal in a digital circuit to verify its logic levels and its proper frequency. Figure 11-5 shows several ways oscilloscopes are used to measure digital signals.



VERTICAL SCALE = 1 VOLT/DIVISION
HORIZONTAL SCALE = 200 ns/DIVISION

BINARY 0 = .2 VOLT

BINARY 1 = 2.8 VOLTS

PERIOD = 600 NANoseconds

FREQUENCY = $1/600 = 1.6667$ MHz

RISE TIME = 80 NANoseconds

Figure 11-5
Measuring logic levels, period, frequency, and rise time with an oscilloscope.

In troubleshooting digital circuits, it is usually common to monitor several different signals simultaneously. In order to determine whether the circuit is functioning properly, one signal in the circuit must be compared to another to verify that certain events are in the proper sequence or at the proper time. For that reason, dual-trace or multitrace oscilloscopes are preferred for digital troubleshooting. Measurements such as propagation delay can only be made on a dual-trace scope, as illustrated in Figure 11-6.

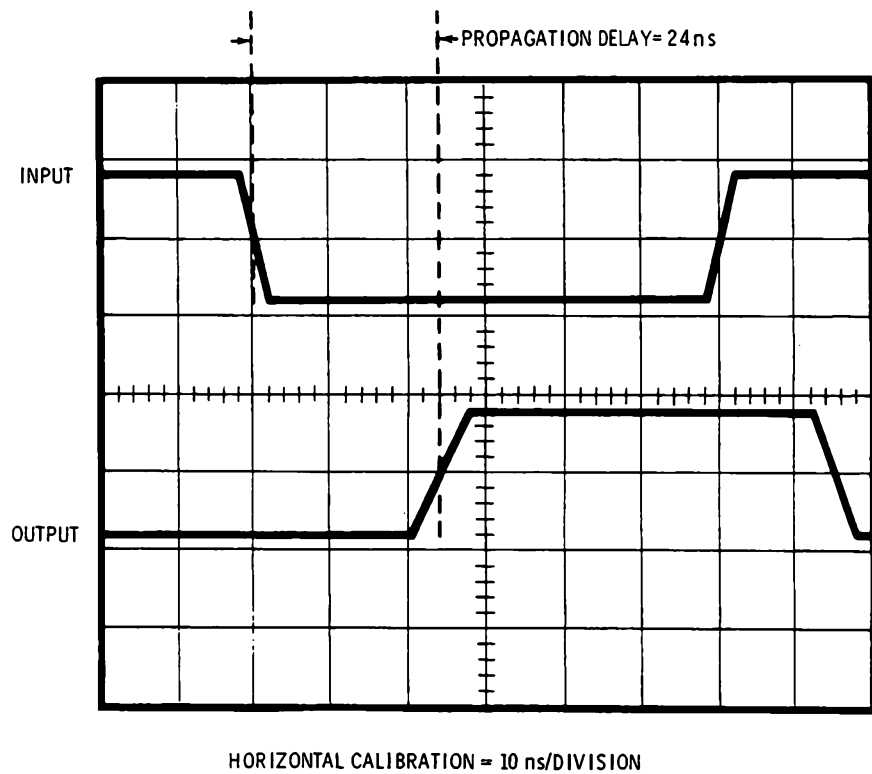


Figure 11-6
Measuring the propagation delay of an inverter on a dual trace oscilloscope.

There are several different kinds of multitrace oscilloscopes available. One kind uses a single electron beam CRT, and multiplexing or alternating sweeps are used to create the effect of 2, 4, or more traces on the screen. The multiplexed or chopped mode of operation gives the effect of displaying multiple channels concurrently, but such schemes are often not usable at the high frequencies encountered in digital equipment. The alternate sweep arrangement is available on some scopes but the time relationship between signals is not faithfully maintained.

The best type of multitrace oscilloscope is one with a dual-beam CRT. Here, two traces can be displayed simultaneously, not just concurrently. Dual-beam oscilloscopes are usually far more expensive, but most of the time, they are absolutely necessary for best testing and troubleshooting.

Oscilloscopes used in digital troubleshooting should also have triggered sweep. Most high quality scopes have this feature.

Triggered sweep means that the horizontal trace on the oscilloscope can be triggered or initiated by an external signal. This can be the signal that is being viewed or some other related signal. Triggered sweep allows you to make more meaningful timing measurements in the digital circuits.

One of the most important characteristics of an oscilloscope for digital testing and troubleshooting is bandwidth. The vertical amplifiers in the oscilloscope must have sufficient bandwidth in order to measure the very high frequencies that usually occur in digital circuits. The bandwidth figure of an oscilloscope is usually the upper frequency of the vertical amplifier. The higher the bandwidth, the more accurately the oscilloscope will display the actual shape of a digital signal. Wide bandwidth is also absolutely essential in making accurate rise and fall time and proper propagation delay measurements.

The bandwidth of the vertical amplifier is often rated in terms of rise time. The rise time (t_r) and the bandwidth (BW) of the vertical amplifier are related by the formulas given below:

$$BW = .35/t_r, \text{ or } t_r = .35/BW$$

In these formulas, BW represents the bandwidth in MHz. The rise time, t_r , is in microseconds. If an oscilloscope has a rise time of 50 nanoseconds (.05 microseconds), its bandwidth is $BW = .35/.05 = 70$ MHz. An oscilloscope with a 50 MHz bandwidth has a rise time of $t_r = .35/50 = .007$ microseconds or 7 nanoseconds. An oscilloscope cannot measure rise/fall times less than its rise time specification. The wider the bandwidth and the smaller the rise time, the higher the quality of the oscilloscope. For most digital measurements, an oscilloscope must have a minimum bandwidth of 10 MHz. High speed TTL circuits are best tested with oscilloscopes in the 35 to 60 MHz range. For very high speed ECL logic, oscilloscopes with bandwidth of 100 MHz to 1 GHz are available. Figure 11-7 shows an oscilloscope designed for digital measurements.

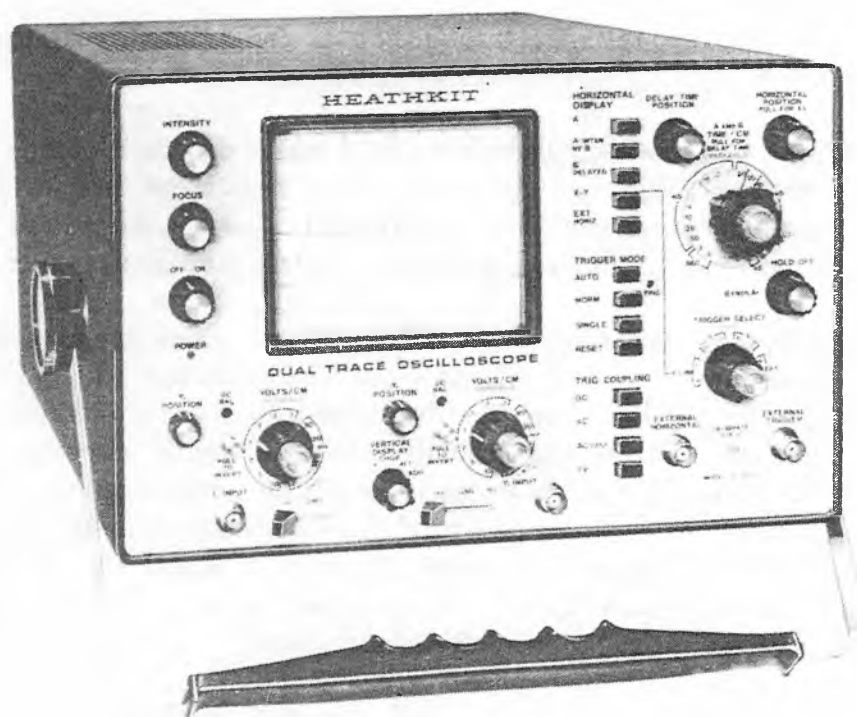


Figure 11-7
A dual beam 35 MHz Oscilloscope.

In addition to being able to measure logic levels as well as time and frequency characteristics, the oscilloscope is particularly valuable in locating noise and distortion problems. Noise is very common in digital circuits and it is difficult to locate. An oscilloscope allows you to spot voltage spikes and glitches that would never show up in measurements with a multimeter. The shape, size, and other characteristics of the noise pulses often give a clue to their origin.

Ringings, shown in Figure 11-8A, is another problem that often causes problems in high speed circuits. Ringing is the high frequency oscillation that often accompanies the leading or trailing edges of a pulse signal. It is often caused by improper grounding in the system.

Distortion of a digital signal is often an indication that a fault exists. Excessive rise and fall times, excessive pulse sloping, or other distortion such as shown in Figure 11-8B, give a very clear indication that a problem exists.

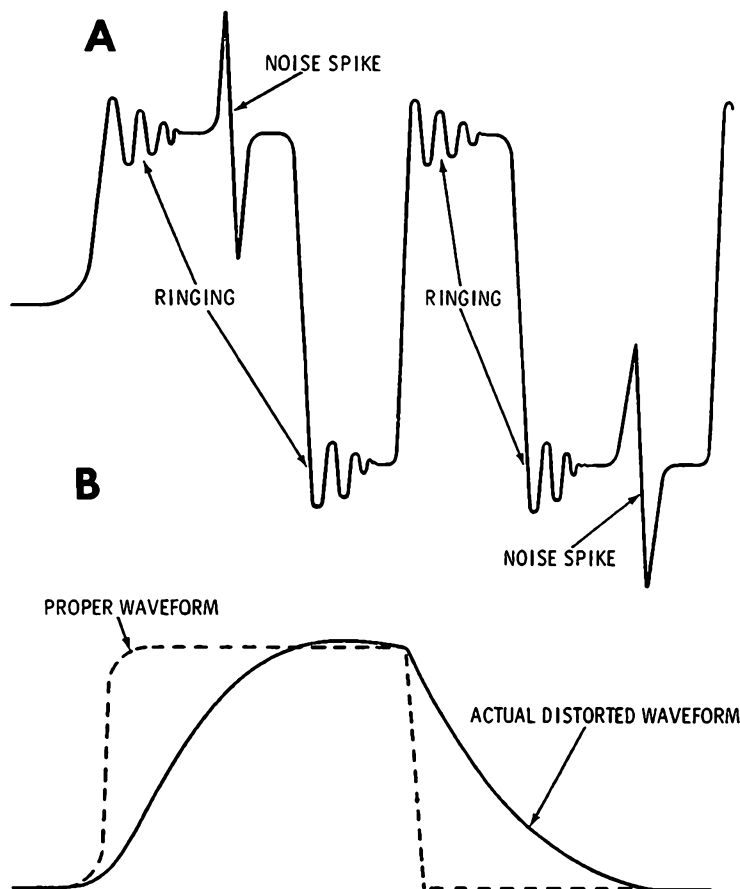


Figure 11-8
Oscilloscopes are good for locating noise and ringing(A) and distortion (B).

A good oscilloscope, like the multimeter, is almost essential to rapid digital troubleshooting. Although they are usually expensive, they will greatly shorten and simplify troubleshooting.

You can use an oscilloscope to measure virtually all common digital signal characteristics, but it is not the best or most accurate instrument to use in all cases. For example, a digital multimeter will give a far more accurate measurement of logic levels than a scope. A standard digital counter will give a more accurate measurement of frequency than a scope. A scope, however, excels at showing signal characteristics such as distortion, noise, and shape factor.

Logic Clips and Monitors

A wide variety of special test instruments have been designed specifically for digital troubleshooting. In the following sections, we will cover a variety of these instruments. The first and the simplest is commonly called a logic clip. This is a device that clips or clamps onto a standard, dual in-line integrated circuit package. Contacts make connection with all of the pins on the IC. The contacts are brought out at the top of the clip so that multimeter and oscilloscope probes can be easily attached. A logic clip is shown in Figure 11-9.

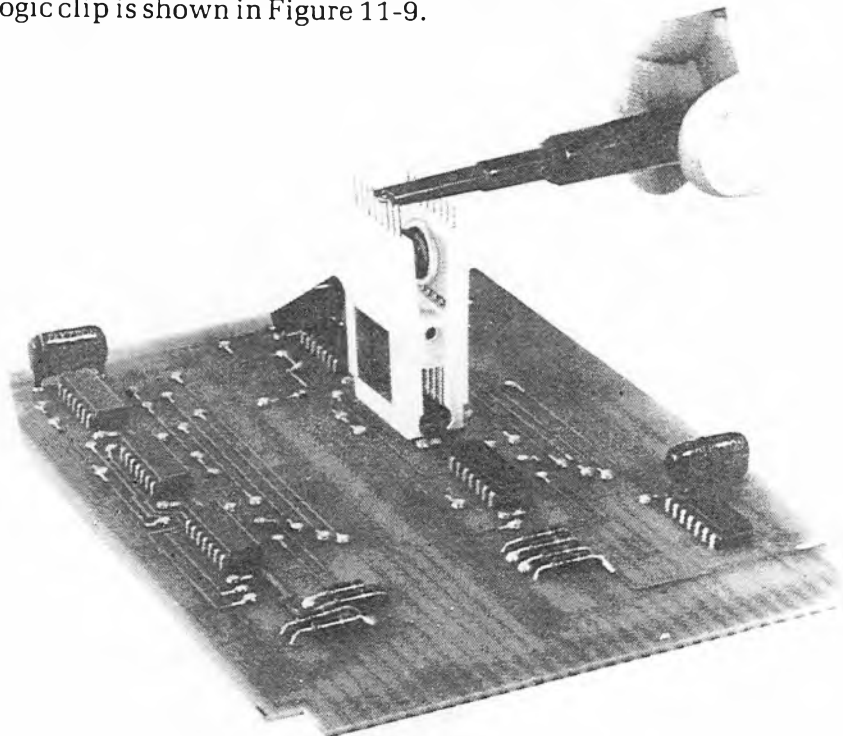


Figure 11-9

A logic clip.

Photo courtesy AP Product.

A logic monitor, shown in Figure 11-10, is similar to a logic clip in that it attaches to an IC DIP. Connected to each pin is an LED used as a logic level indicator. The result is a visual display of the binary state of each pin on the IC. The LED indicates the existence of a binary 0 (off) or a binary 1 (on). You can check for binary 0 and binary 1 states on an IC with a multimeter, of course. But this is very tedious and time consuming and you will need to write down the states on each pin in order to remember them. With a logic monitor you are able to observe every state on the IC pin simultaneously. This gives you a very quick check of their states to determine if a problem exists or not.

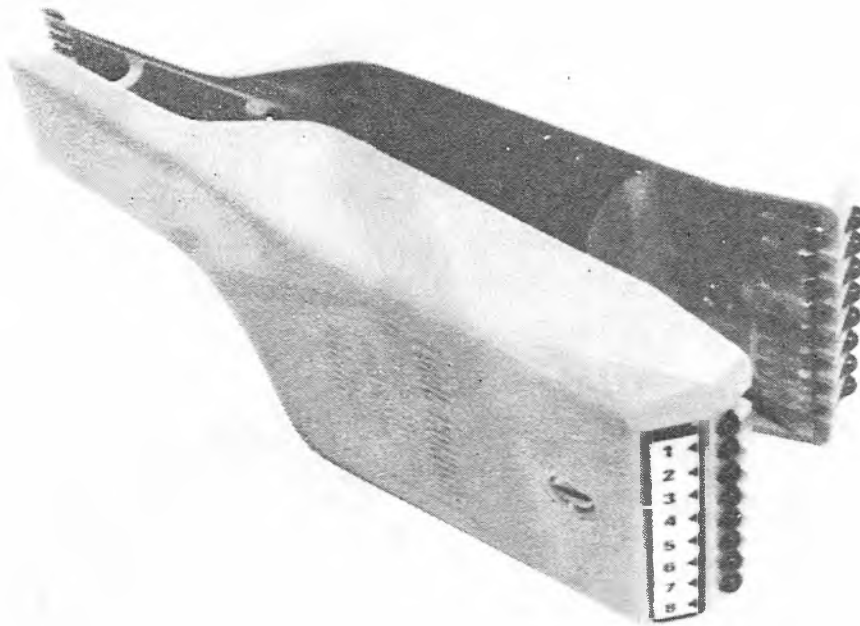


Figure 11-10
A logic monitor.
Photo courtesy Global Specialties.

A logic monitor is a great time saver in determining faults in a digital circuit. Keep in mind, however, its primary value is in static testing. Static testing infers that a digital circuit is not operating at high speed under the control of the clock. It means that the logic states are fixed or stable. Despite this limitation, logic clips are widely used because they greatly speed up and simplify the troubleshooting process.

Logic Probe

A logic probe is a specialized digital test instrument designed to replace the multimeter for static testing and, in some instances, the oscilloscope for dynamic testing. A logic probe is an instrument used to indicate the logic status at a single point in a digital circuit. Circuitry within the logic probe examines the voltage at the point of measurement and determines whether it is a binary 0 or binary 1. The probe then gives an appropriate output on an indicator light. Usually the logic probe circuitry draws its power from the power supply of the equipment under test. A typical logic probe is shown in Figure 11-11.

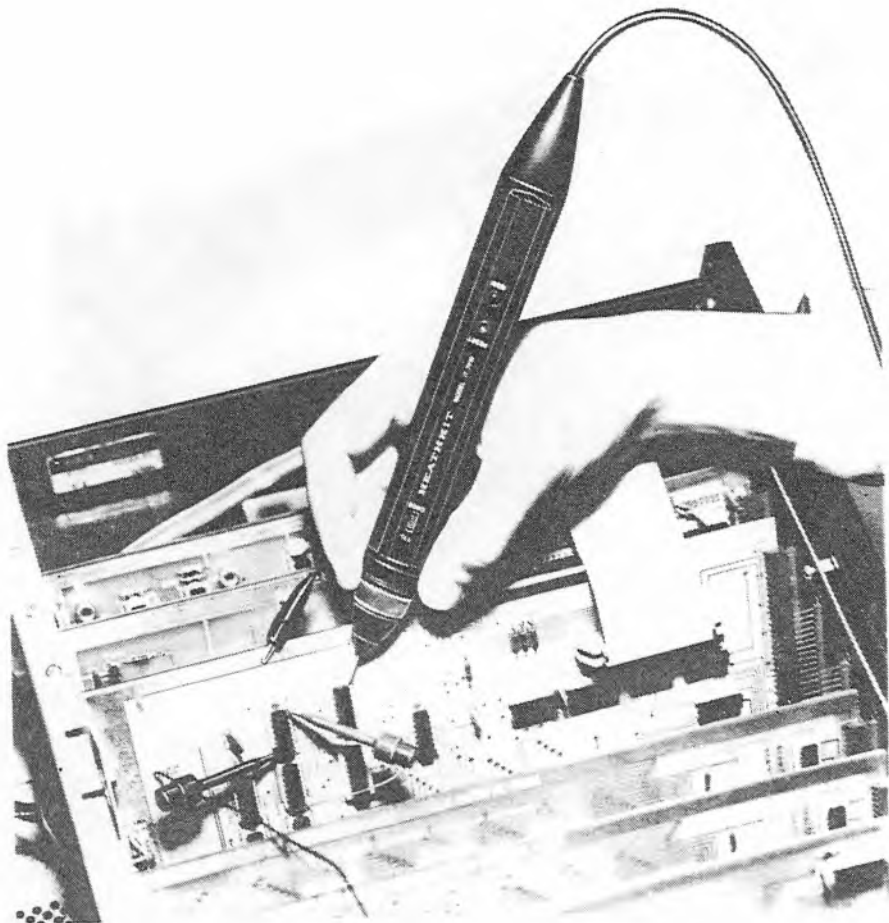


Figure 11-11
A logic probe.

Most logic probes have one or more indicator lights. These are usually incandescent lamps or LEDs. On very simple logic probes, only a single indicator is used. When the indicator is off a binary 0 is indicated and when the indicator is on, a binary 1 is indicated. If two indicator lamps are used, one designates a binary 0 and the other designates a binary 1. For example, if the logic probe is touched to a point that is a binary 0, the binary 1 indicator lamp will be off while the binary 0 indicator lamp will be on. Touching the probe to a point in the circuit that is a binary 1 will cause the binary 0 indicator to be off and the binary 1 indicator to be on. This basic idea is illustrated in Figure 11-12.

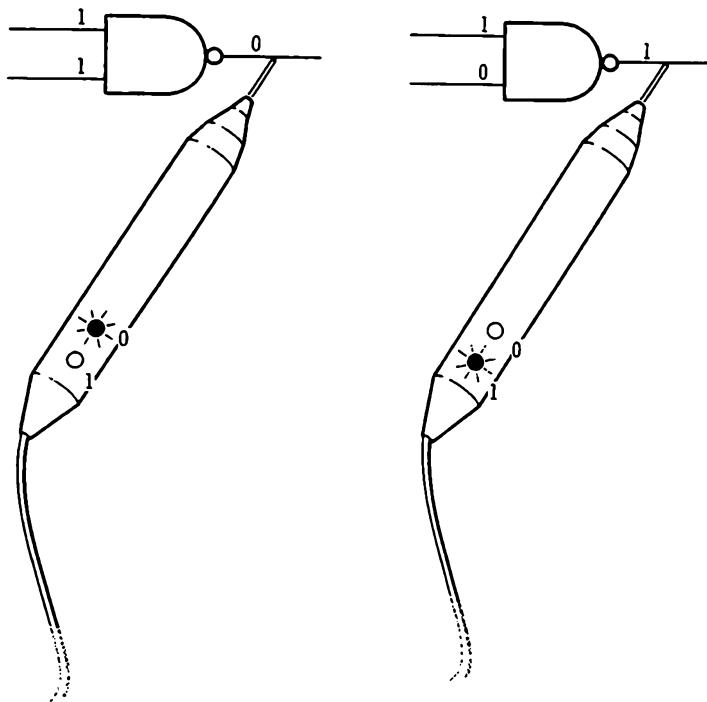


Figure 11-12

Using a logic probe with 0 and 1 output indicators.

Since there are different thresholds for binary 0 and binary 1 levels in the various kinds of logic circuits, the probe will have to be set up to detect them. Some manufacturers have different probes for each of the various kinds of logic circuits (TTL, CMOS, ECL, etc.). Other logic probes contain circuitry for detecting logic levels of several different types of circuits. A selector switch is used to enable the appropriate threshold detection circuitry.

In addition to being able to test for binary 0 and binary 1 levels, a logic probe will also indicate open circuits and invalid logic levels. For example, a disconnected TTL input will measure approximately +1.5 volts. This is an invalid TTL level between the normal binary 0 and binary 1 states. If an invalid level such as this is detected, the probe indicators may be off or may simply glow dimly. The actual indication depends upon the type of probe and manufacturer. It is essential that you read the instruction manual for your probe before using it to be sure that you understand the rules for the various indications.

While logic probes are most often used for static logic checking, they can also be used to perform dynamic tests as well. High frequency repetitive signals are indicated by the flashing of the logic probe indicator lights. Most logic probes can detect periodic signals with frequencies as high as 100 MHz. Of course, the indicator lights will not follow at these speeds; they will usually have a dim glow to indicate a high speed logic signal. Most logic probes have internal circuitry that is used to flash the indicator lights off and on at a 5 to 10 Hz rate if the probe is connected to a high speed repetitive signal. Again the exact indication depends upon how the logic probe works.

Another feature usually indicated in the better logic probes is a memory circuit. The memory is usually a flip-flop or an SCR (thyristor). The memory circuit is used to detect the occurrence of single pulses or logic level transitions. A manual reset switch on the probe is used to reset the flip-flop or turn off the SCR. When the single pulse occurs, the flip-flop is set or the SCR is turned on. This state is usually indicated by a separate indicator light. Most logic probes with a memory circuit can detect single pulse widths as narrow as 10 nanoseconds.

The logic probe is nearly a complete replacement for a multimeter in most static logic tests. A multimeter is essential only when you must know the exact value of a voltage.

The logic probe is not a perfect replacement for the oscilloscope in dynamic testing. The oscilloscope is best because it shows so much detail. However, for simple tests, the logic probe is a useful substitution in many applications. For example, you can make simple tests such as determining whether or not the clock is running in a piece of digital equipment with a logic probe. Logic probes are smaller, less expensive, and more portable than oscilloscopes and are ideal for field service. You can perform a high percentage of tests with a logic probe, but if you encounter unusual difficulty, you will need an oscilloscope for additional information.

Examining the operation of a typical logic probe will give you a better idea about its application. Consider the Heathkit Model IT-7410 Logic Probe. This instrument is designed for testing both TTL and CMOS logic circuits. It has an input impedance of 400k ohms. With its separate threshold detection circuitry, it can detect the standard TTL levels of CMOS levels. The probe has both binary 0 and binary 1 indicator lamps, and can detect square waves up to 80 MHz for CMOS circuits and 100 MHz for TTL circuits. The probe will also recognize single pulses or pulse trains with a minimum width of 10 nanoseconds. A memory indicator light turns on for any change in either logic level, and you can turn the memory light off with a manual reset button. Also, the probe can operate from any voltage between 5 and 15 volts.

Figure 11-13 is an illustration of the IT-7410 Logic Probe. The most important features including indicators and controls are explained. The logic level indicators are located in the housing near the probe tip. A red indicator is used to designate the binary 1 state, while a white indicator is the binary 0 state. If an invalid logic level between binary 0 and binary 1 is detected, indicator lights may be off, overly bright, or glowing dimly. For repetitive input signals, both indicators flash alternately at about a 5 Hz rate.

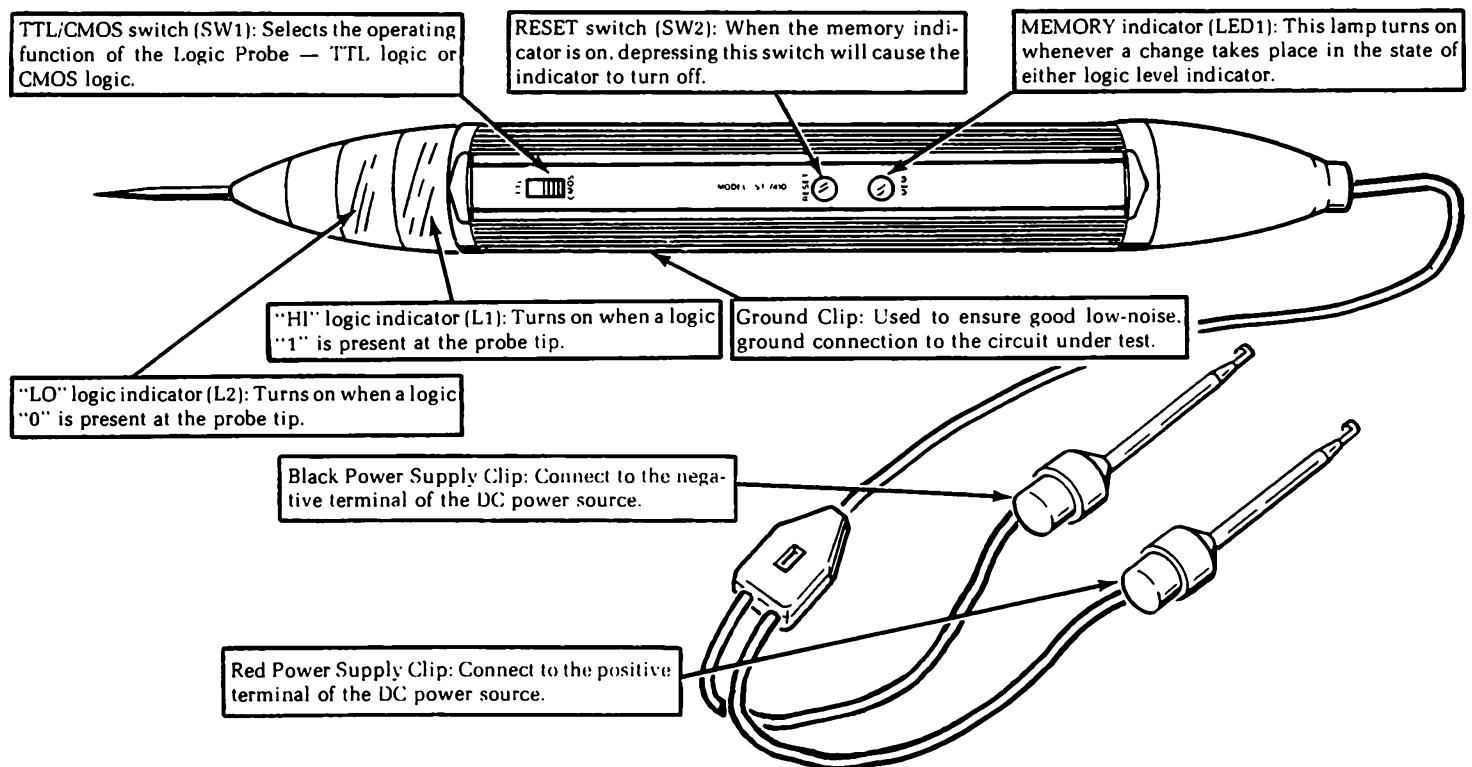


Figure 11-13
Heathkit IT-7410 logic probe.

A memory indicator LED turns on when a logic level change takes place. If the voltage at the point in the circuit where the probe is touched changes from binary 0 to binary 1 or from binary 1 to binary 0, the memory indicator will light and remain on. A memory reset pushbutton is used to clear or reset the memory circuit.

This probe will measure either TTL or CMOS logic levels. The internal thresholds are set to register binary 0 and binary 1 for the logic levels given below:

TTL binary 0 0 — +.8 volts

TTL binary 1 +2.1 — +5 volts

CMOS binary 0 30% of supply voltage

CMOS binary 1 70% of supply voltage

Figure 11-14 shows the typical inputs that can occur in testing. The logic probe response is given for each.

Performance Limits


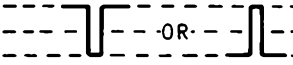
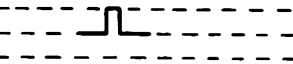
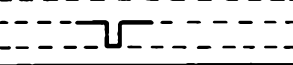




INPUT CONDITION	LOGIC LEVEL INDICATORS		RESPONSE LIMITS		
	HIGH (RED)	LOW (WHITE)	TTL @ 5.0 VDC $V_H = 2.4; V_L = 0.4$	CMOS @ 5.0 VDC $V_H = 4.5; V_L = 0.5$	CMOS @ 15 VDC $V_H = 13.5; V_L = 1.5$
V_H ----- OPEN----- V_L -----	OFF	OFF	—	—	—
----- ----- -----	ON	OFF	—	—	—
----- ----- -----	OFF	ON	—	—	—
	FLASH	FLASH	100 MHz* maximum	100 MHz* maximum	80 MHz* maximum
	FLASH	FLASH	10 ns* minimum	10 ns* minimum	10 ns* minimum
	FLASH	OFF	10 ns* minimum (PRF ≤ 1.0 MHz)	10 ns* minimum (PRF ≤ 600 kHz)	10 ns* minimum (PRF ≤ 600 kHz)
	OFF	FLASH			
	FLASH	OFF	600 kHz maximum	250 kHz maximum	250 kHz maximum
	OFF	FLASH			
	FLASH	OFF	1.0 μs minimum (PRF ≤ 600 kHz)	2.0 μs minimum (PRF ≤ 250 kHz)	2.0 μs minimum (PRF ≤ 250 kHz)
	OFF	FLASH			
*USE OF HIGH FREQUENCY GROUND CLIP REQUIRED.					

Figure 11-14
Typical logic probe inputs and responses.

Figure 11-15 is a general block diagram of the logic probe. The signal being measured is applied to the input filter and protection circuits. This circuit essentially filters out all low frequency signals and, at the same time, protects the internal circuitry from excessive input voltages. Any input voltage over 50 volts DC or 124 volts AC will damage the probe. Internal clamping diodes protect the probe circuitry from accidental overloads.

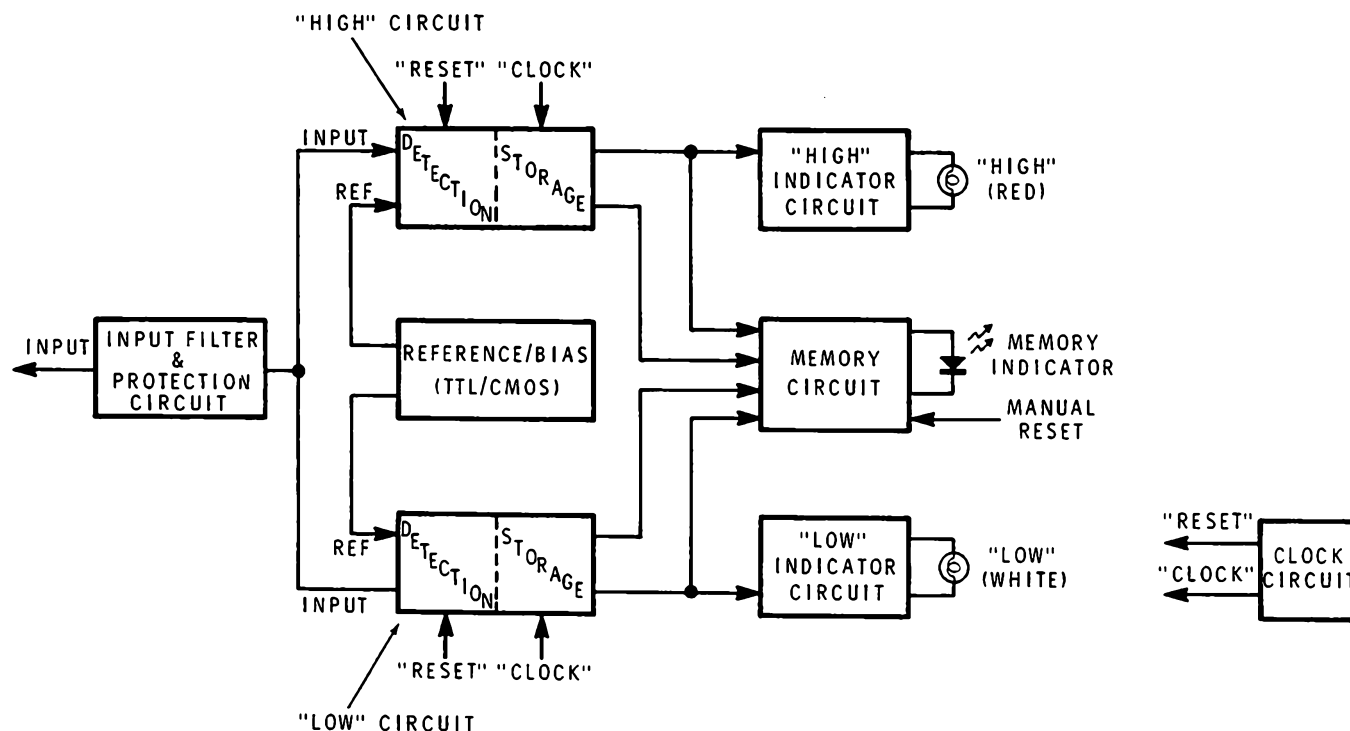


Figure 11-15
Block diagram of IT-7410 logic probe.

The output of the input filter and protection circuit is fed to two detection circuits simultaneously. The detection circuits are voltage comparators, one for TTL inputs and the other for CMOS inputs. Each comparator compares the input to fixed reference voltages which are determined by the TTL or CMOS thresholds values. The output of the detection comparators feed storage circuits. The storage circuits are made up of a number of flip-flops that store the binary 0 or binary 1 state detected. The storage circuits operate the high and low indicator circuits.

A clock circuit generates the timing pulses for the logic probe. The clock runs at a 10 Hz rate. It first generates a reset pulse that clears the storage circuits, then 100 milliseconds after the reset pulse, a clock pulse occurs. The clock pulse causes the input state detected to be stored and the appropriate indicator light to turn on. This reset/clock pulse sequence continues repetitively as the input signal is monitored.

Finally, a memory circuit made up of an SCR is used to determine when a logic level change occurs. Any high-to-low or low-to-high input state change causes the memory SCR to turn on and the appropriate indicator LED to light.

Figure 11-16 shows additional detail on the various types of input signals that the logic probe deals with and the resulting logic probe indications.

A logic probe is one of the most useful test instruments available for testing digital circuits. Like most test instruments, they require a little practice to use effectively.

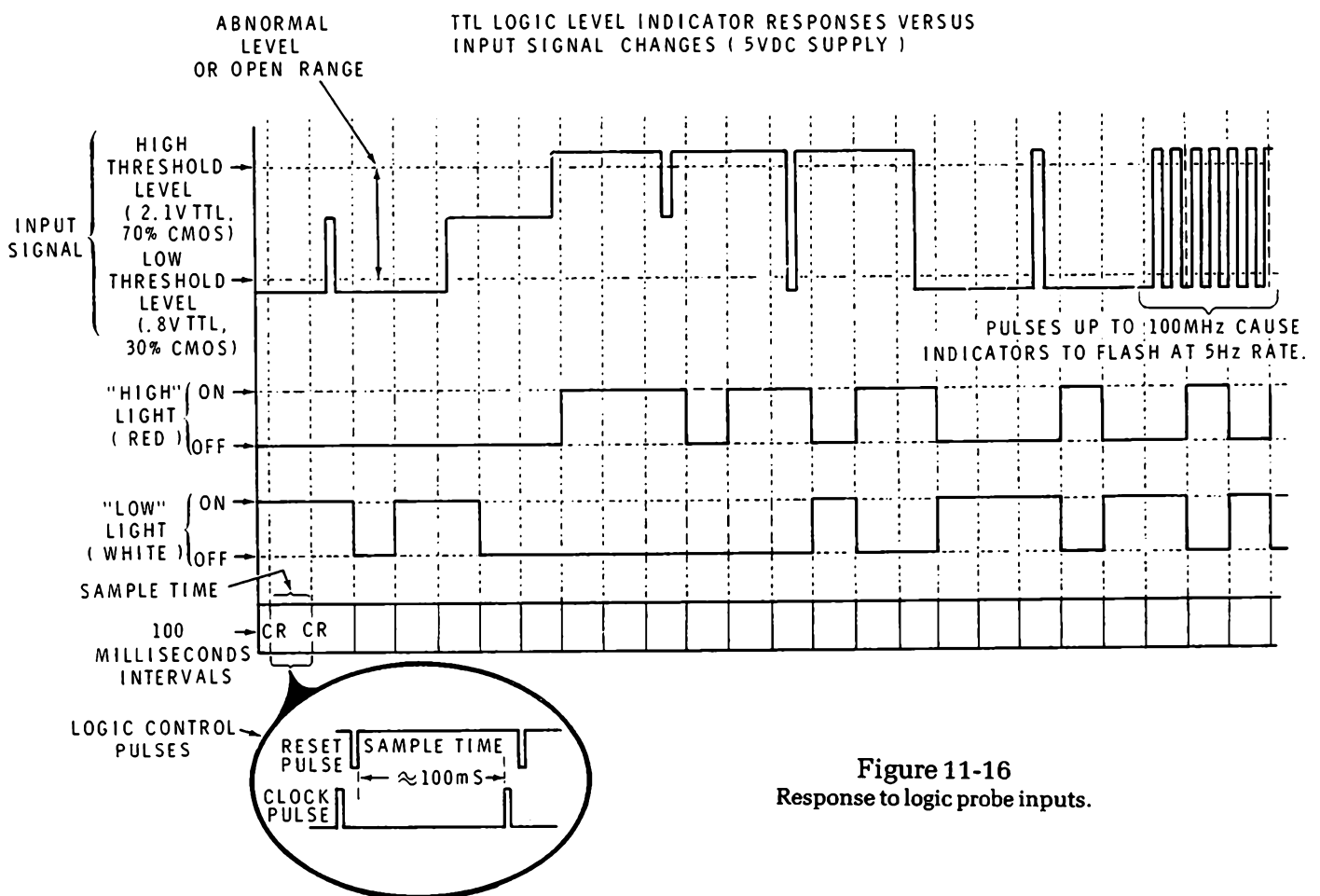


Figure 11-16
Response to logic probe inputs.

Logic Pulser

When you test digital logic circuits, it is often necessary to trigger a gate, flip-flop, or other circuit with a logic pulse to see if it is operating properly. One way to do this is to remove the integrated circuit from the equipment and test it on a breadboard or in a special integrated circuit tester. Many times, such a tester is not available. Further, the integrated circuit is usually soldered to a PC board and is difficult to remove. For such cases, you can use a special test instrument called a “logic pulser.”

A logic pulser is a special form of signal generator that produces narrow logic pulses that can be applied to logic circuits to test them. Typically, the logic pulser is housed in a package that is similar to that of the logic probe. To use the logic pulser, you touch the point or probe end to the circuit input or output and press a button to generate a pulse. The logic pulser is usually used in conjunction with a logic probe. When you press the pulser button, one pulse is generated. The logic probe is used to monitor the circuit under test and to note any logic level changes as a result of the pulse applied. Figure 11-17 shows how a logic pulser is used.

The logic pulser contains a circuit that automatically determines which polarity of pulse to generate. For example, when you touch the logic pulser to the point in the circuit to which the pulse is to be applied, the internal sensing circuit determines whether that point in the circuit is a binary 0 or binary 1. If that point in the circuit is a binary 0, it will be driven to the binary 1 level when you press the pulser button. If the pulser circuitry detects a binary 1 level, that point in the circuit will be driven to a binary 0 when you press the pulser button.

You can apply the logic pulser to virtually any point; you can use it at both the inputs and outputs of the logic circuit. It will either sink current (act as a load) or provide source current, which allows it to change the logic state at any point, and its high fan-out capability allows it to override any other circuit connections.

This ability to change the state of any point in the circuit is extremely valuable because it permits complete in-circuit testing. You can check ICs and complete digital circuits without removing the ICs or making other circuit changes. With some test equipment, you must sometimes break (open) a circuit at certain points or remove components in order to do the necessary testing; with a logic pulser, it isn't necessary.

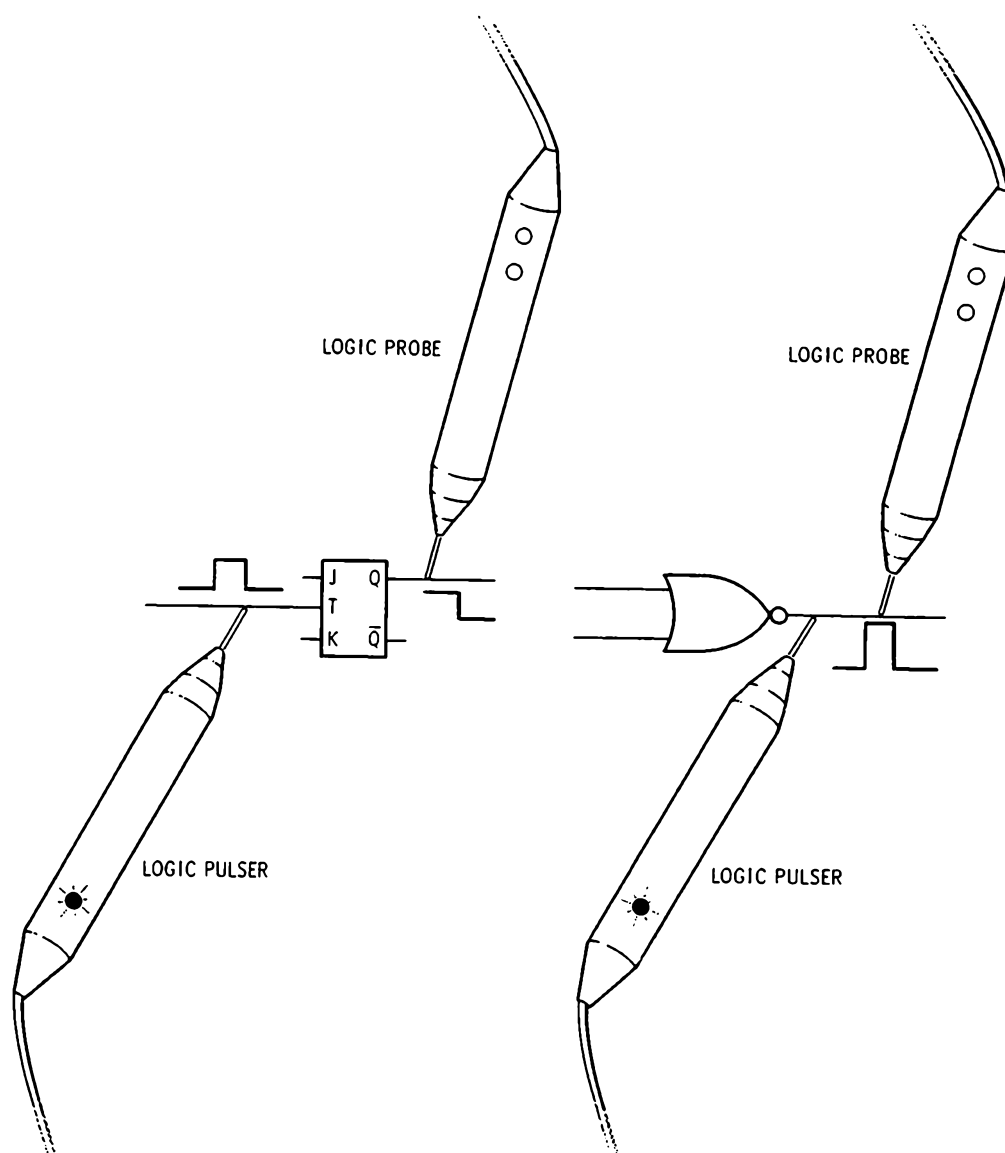


Figure 11-17
Using a logic pulser and a logic probe together to test circuits.

Pulsers are available for the various types of logic. Different amplitudes and drive capabilities are required for TTL, CMOS, ECL, and other logic circuit types.

Normally when you press the logic pulser trigger button, the pulser will automatically generate a single pulse. A typical pulse width for a TTL pulser is 300 nanoseconds to 1 microsecond. A 10-microsecond wide pulse is typical for a CMOS probe.

A logic pulser can also generate multiple pulses. When you hold down the trigger button for longer than one second, the pulser begins to generate a series of off/on pulses at approximately a 100 Hz rate. Such pulse bursts are useful for triggering logic circuitry several times to determine whether it is operating properly.

Current Tracer

A current tracer is another type of portable digital test instrument. Like the logic probe and the logic pulser, the current tracer is housed in a hand-held probe. Where the logic probe is used for detecting voltage levels, the current tracer measures the relative value of the current in a conductor. A special probe at the end of the current tracer senses the magnetic field generated by a current flowing in a wire or a connection on a printed circuit board. In most applications, a logic pulser is used to supply a pulse to the circuit. The current tracer is used to detect that pulse current, or other current pulses in the circuit, generated as a result of the stimulus given by the logic pulser.

The presence of a current is indicated by a lamp in the current tracer housing. The current tracer usually has a sensitivity adjustment so it can sense any current from approximately 1 milliampere to 1 ampere. When the sensitivity level is set, the probe will indicate the presence of a current in excess of the sensitivity level threshold. Like the logic probe and current pulser, the current tracer usually takes its operation power from the supply of the equipment under test.

The current tracer is particularly useful in troubleshooting low impedance circuits. The current tracer is ideal for locating short circuits and for troubleshooting data buses and wired OR circuits. The current tracer can locate shorted IC inputs and outputs, solder bridges on printed circuit boards, shorts in power supplies, and shorted conductors in cables.

The current tracer is also very useful in dealing with data buses. Typically data buses are multiplexed, and many inputs and outputs are connected to common lines. It is not easy to find problems in such circuits unless you disconnect the various circuits from one another. As indicated earlier, this is difficult, if not impossible, in most normal troubleshooting.

Wired OR circuits are similar in that many outputs are connected together. Whenever a problem exists in one circuit, it affects all of the others. Locating the defective circuit is difficult with voltage tracing techniques unless the individual circuits can be disconnected from one another. In all of these cases, the current tracer provides an easy way to locate the trouble without disconnecting any of the circuitry.

The basic principle of operation of the current tracer is that the circuit driving a low impedance fault or short must be delivering the majority of the current. The fault can be located by tracing the path of this high current.

To find a short with a current tracer, you touch the probe tip to the wire or printed circuit trace near the expected short, then adjust the sensitivity control until the indicator lights. Next, move the current tracer along the wire or printed circuit connection. Follow all paths connected to this point, and watch the indicator. Usually, you will locate the driving circuit, as well as the short.

Figure 11-18 shows an example of how you might use a current tracer to detect a short circuit. The solder bridge in the circuit has caused the output of two logic gates to be shorted together. You touch current tracer tip to the output of gate A and adjust the sensitivity level so that the indicator lights. Then follow along the path where the current keeps the indicator on. The indicator will remain lighted until it passes the solder bridge. This means that the current tracer has lost the main current path and has found a lower current path. At the point in the circuit where the indicator goes off, you will find the solder bridge. Visually inspecting the printed circuit board will usually turn up the problem.

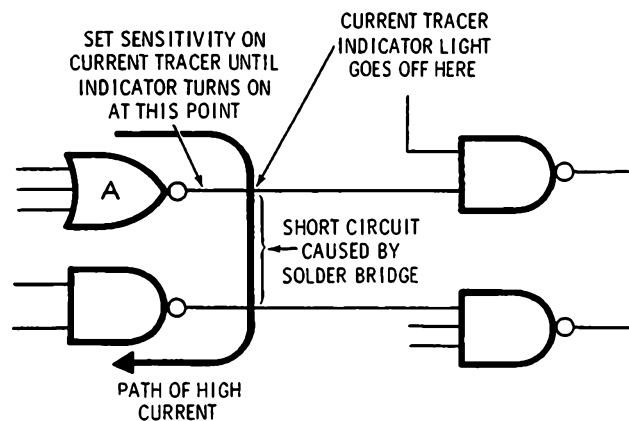


Figure 11-18
Using a current tracer to locate a short circuit.

Figure 11-19 is an example of how you can use the current tracer to locate a shorted gate output in a wired AND circuit. Initially, you connect the current tracer to the pull-up resistor and adjust its sensitivity so the indicator is on. Then use the current tracer to follow the current to each of the gate outputs. As you move the current tracer along the current path, the indicator light will go out on those gates not causing the problem. The current tracer indicator will stay on at the output of the shorted gate. You can use a logic pulser, if necessary, to generate the test current in the circuit.

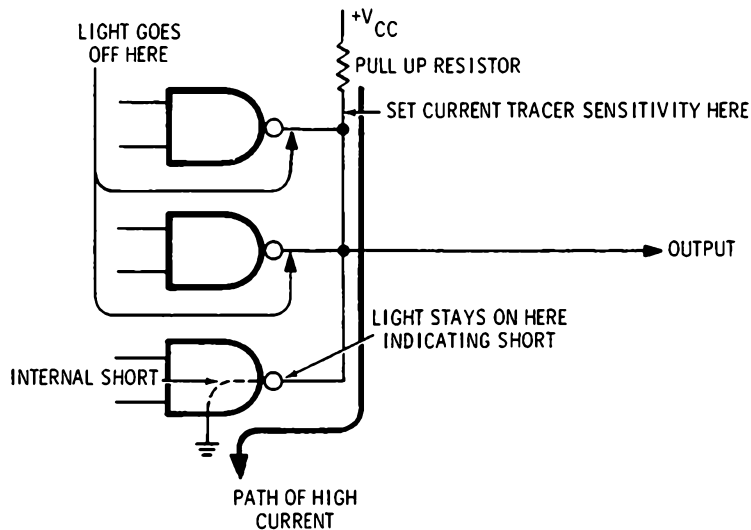


Figure 11-19
Using a current tracer to troubleshoot wired
AND circuits.

Logic Analyzers

One of the difficulties you will often encounter in troubleshooting digital equipment is that, in order to determine if a circuit is operating properly, you must monitor and compare many different logic signals simultaneously. The proper operation of most digital circuits depends upon the timing, sequencing, and relationships between a number of different signals. Logic probes, current tracers, multimeters, and other such instruments are capable of monitoring only one point in a circuit at a time. This is OK for simple troubleshooting, but for complex systems it is sometimes inadequate.

You can use oscilloscopes to monitor multiple points in a circuit, but most oscilloscopes are limited to two traces, or four at the most. This is still inadequate for many complex systems. For multiple tracings, you can use the chopped or alternate modes of sweep, but the alternate mode, while creating the effect of multiple channels, does not always display the multiple inputs with the correct timing relationships. In the chopped mode, the frequency of the signals being monitored is severely limited. As a result, even multitrace oscilloscopes are inadequate for diagnosing complex digital problems.

Another factor is that digital systems have become more complex over the years. With the addition of microprocessors, semiconductor memories, and a variety of other LSI and VLSI circuits, digital equipment, although small in size, can be incredibly complex. Combine this with high speed and you have one of the most difficult and challenging troubleshooting problems.

To cope with such complexities, special digital test instruments have been developed. The most important of these is the logic analyzer.

A logic analyzer is essentially a recording device that accepts multiple input channels of digital data. Each input channel accepts a digital signal from a single source in the equipment under test. Typical logic analyzers have from 8 to 16 input channels, although logic analyzers are available with up to 64 inputs, and can typically accommodate data rates of 50 to 100 MHz.

The multiple input signals are sampled and stored in a semiconductor memory built into the logic analyzer. Since the logic input signals are nothing more than binary data, a standard RAM can be used to store the various input states. All of the input signals occurring simultaneously are sampled and stored together within the computer memory for a fixed duration of time.

Once the input signals are sampled and stored, they can be displayed on a standard CRT. A variety of different display modes can be selected. For example, a pulse train or time display mode allows all input signals to be displayed as timing waveforms on the CRT. In this way, the logic analyzer acts as a multi-input digital oscilloscope.

The data in the memory can also be displayed in a binary format. Instead of timing pulses, only 1's and 0's are written on the CRT display. And finally, a map or graphic format display allows the data to create a unique pattern on the screen that can be used to recognize proper and improper operation. We will discuss these display modes in more detail later.

A simplified block diagram of a logic analyzer is shown in Figure 11-20. The multiple input lines are attached to the equipment under test by small clips. The logic input signals are then fed to input signal conditioning circuitry. The probes are like scope probes, and the input circuitry is similar to the logic threshold detection circuitry in a logic probe. In order to accurately detect, store, and display digital signals, the input signal conditioning circuitry must sense the upper and lower logic levels appropriate for the circuitry under test. This may be TTL, CMOS, ECL, NMOS, or some other logic type.

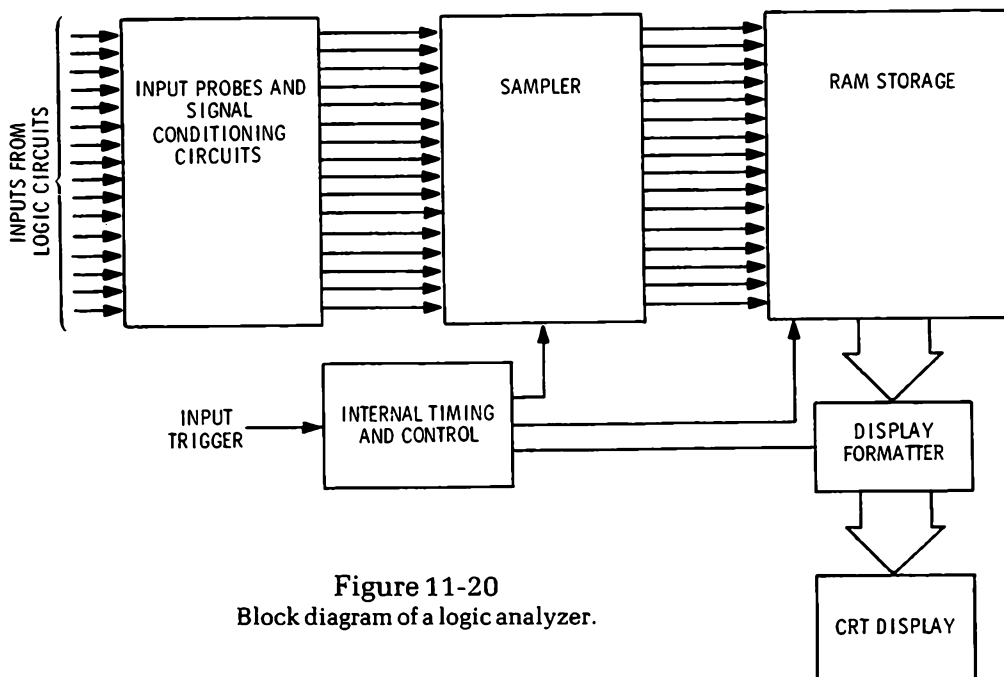


Figure 11-20
Block diagram of a logic analyzer.

Once the input signals have been properly sensed and conditioned, they are fed to a sampler circuit. The sampler is essentially a set of switches or gates that are enabled for a short period of time to let the input signal pass. The sampling time interval is only a few nanoseconds in most high speed logic analyzers. This is a sufficient duration of time to sample the logic level. All sampler switches and gates are operated simultaneously so that the samples of the input signals are taken at the same time.

The sampled signals create a multibit binary word. For 16 input channels as shown, a 16-bit binary word appears at the output of the sampler. This 16-bit word is stored in a semiconductor memory made up of standard MOS RAM chips. Its size depends upon the number of input channels and the amount of data to be stored. The RAM in most logic analyzers is capable of storing several hundred to several thousand of these multibit input words.

The internal timing circuits control the sampling and storing of the input signals. These are in turn typically triggered by an external input signal. Usually, the clock signal from the circuit under test is used to trigger the sampling and memory circuits. A signal derived from the clock, or another input signal which may be used as a reference, can also be used to trigger the sampling and storage operations. Most logic analyzers also contain an internal clock circuit which may be used to handle the timing functions. Some logic analyzers have a front panel pushbutton for manual triggering, which initiates one sampling and storage operation.

Once the input signals have been sampled and stored, they are displayed in one of three forms on the CRT. The three display modes are: the timing mode, the data mode, and the map mode.

TIMING MODE

The timing mode or pulse train format display is similar to that of a standard oscilloscope. The signals stored in memory are translated into standard voltage timing waveforms which are displayed simultaneously on the CRT as in Figure 11-21.



Figure 11-21
Model D132 Logic Analyzer showing timing mode
display.
Photo courtesy Intech.

This mode is best suited for finding timing faults. All input channels were recorded simultaneously and are displayed in the same way. You can easily see the timing relationship of various pulses. The display shows the logic voltage levels with respect to time. In this mode, the logic analyzer is simply a high resolution multichannel oscilloscope. Most logic analyzers can display up to 16 input channels simultaneously.

DATA MODE

The data mode or binary format display is simply a listing of the memory contents displayed as binary 1's and 0's on the CRT. Instead of showing voltage pulses with respect to time, the CRT simply shows a string of binary 1's and 0's. A character generator in the CRT formatting circuitry translates the memory contents into binary 0 and 1 characters.

The data format display is useful in troubleshooting primarily because most operations in a digital system take place on data words. These are usually binary words that are some multiple of 4 or 8 bits in length. Often, it is easier to recognize a fault when it is displayed as a binary number rather than a voltage waveform. If the value of the correct binary word is known, it can be quickly located on a display of binary 1's and 0's.

Figure 11-22 shows the relationship between the time and data mode displays. The waveforms shown are those that might occur at the four outputs of a BCD counter. Note that the binary 0 or 1 values are shown in relationship to the timing waveforms. In the data mode, only the binary 1's and 0's are displayed.

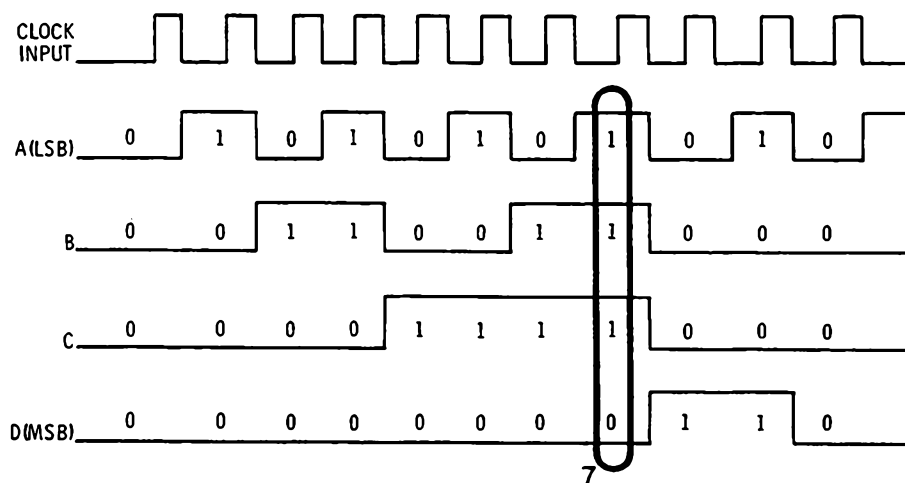


Figure 11-22

Input and output waveforms of a BCD counter as they would be displayed on a logic analyzer: waveforms in timing mode, 1's and 0's in data mode.

Note that in a data mode display, each horizontal row of binary numbers represents one data input channel. To detect a particular binary number or word, vertical numbers are searched. Figure 11-22 shows how the 4-bit code 0111 (representing 7) would appear. A logic analyzer showing a data mode display appears in Figure 11-23. In addition to binary numbers, most logic analyzers also have octal, hex and even ASCII options for the data display mode.

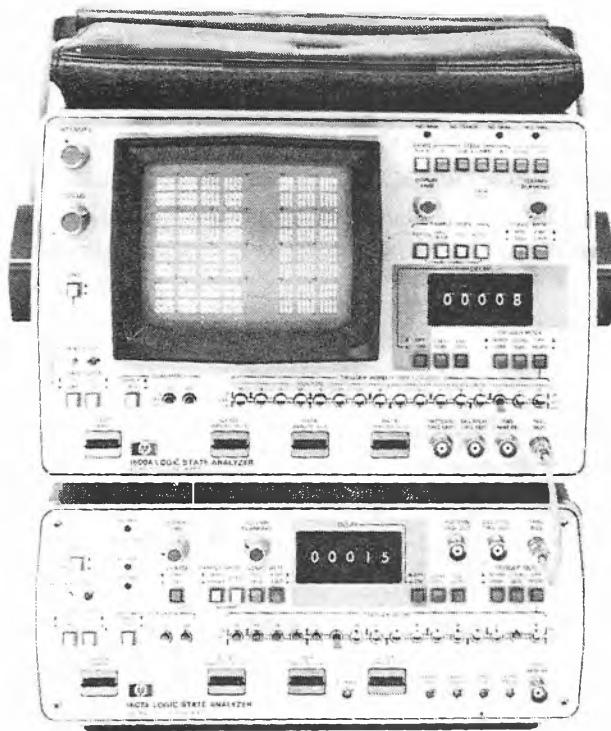


Figure 11-23
Logic analyzer showing a binary data mode display.
Photo courtesy Hewlett-Packard.

MAP MODE

The map mode gives a graphic display on the CRT unique to the input signals. What happens is that the parallel data words are read out of memory into two digital-to-analog converters. Half of the bits of the data word feed one D-to-A converter and the other half of the data bits feed another D-to-A converter. The outputs of these D-to-A converters form X and Y axis sweep voltages which are applied to the horizontal and vertical plates of the CRT.

With this approach, the data read out of the memory into the display causes a unique pattern or signature to be drawn. A light dot generated by the CRT is deflected horizontally and vertically by the data being read out of the memory. As data changes, the position of the light dot on the CRT will jump rapidly from one position to the other. The unique pattern created by a repetitive series of data is easy to recognize. This is shown in Figure 11-24. If the circuit is operating properly, a specific pattern will be obtained. Defects in the circuit will show up as changes in the map or pattern.

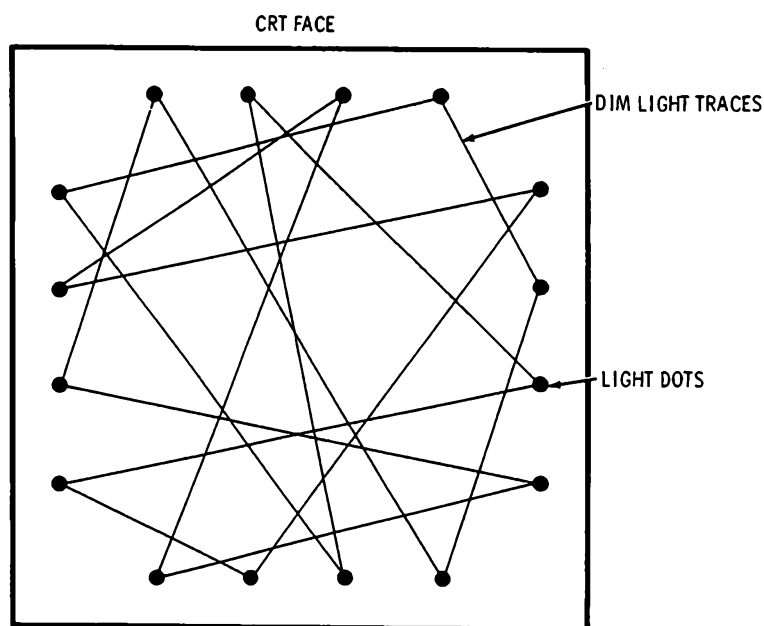


Figure 11-24
Map mode display.

It is important to point out that a logic analyzer is simply a recorder and displayer of data. The term “analyzer” is perhaps misused. The analysis of the information selected and displayed is still subject to interpretation. The human operator must still observe the display and determine for himself whether or not a problem exists. He can usually determine whether the equipment is operating properly from instruction manuals, timing displays, and data word patterns. For new equipment being tested, the design specifications provide the reference information in which the display is compared.

The logic analyzer is a powerful and useful device. It is most helpful in troubleshooting complex digital systems. Special logic analyzers have been built to help locate problems in unique applications. For example, the logic analyzer in Figure 11-25 is used to locate problems in data communications networks using the EIA RS-232 interface.

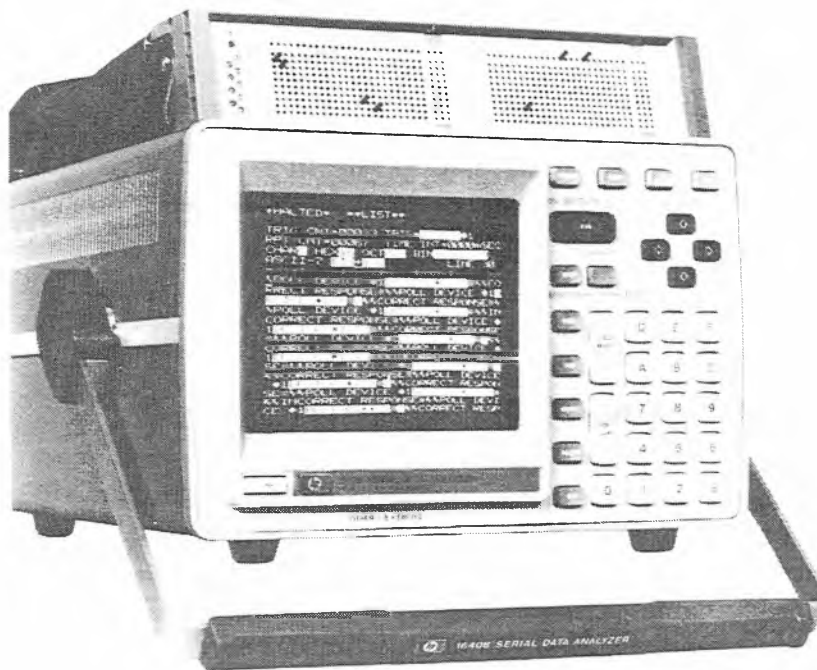


Figure 11-25
A serial data logic analyzer used to find problems in
a data communications network.
Photo courtesy Hewlett-Packard.

Microprocessor equipment is particularly difficult to troubleshoot. Logic analyzers are a must in troubleshooting and repairing microprocessor-based equipment. Special forms of logic analyzers are available to work with microprocessor-based systems. The unit in Figure 11-26 can monitor microprocessor data and address buses and identify specific instruments for display. Binary, octal, and ASCII display modes are available.



Figure 11-26
A Logic analyzer for microprocessor based system
troubleshooting.
Photo courtesy Hewlett-Packard.

Signature Analyzer

A signature analyzer is a special form of digital test instrument that performs a function similar to a logic analyzer. Whereas a logic analyzer monitors many digital signal sources in parallel, the signature analyzer monitors only one point in a digital circuit at a time. The serial pattern of binary 0's and 1's generated at a particular point in the circuit forms a unique signature. If the circuit is operating properly, the serial pattern of binary 0's and 1's will be known. If a defect occurs, the pattern of binary 1's and 0's, and therefore the signature, will be different. This is a simple way to quickly locate faults in digital equipment. The signature itself is usually a 4-digit hexadecimal number.

To use a signature analyzer in locating defective circuits, you must know the correct signatures for each point in the circuit. For many types of digital equipment, the manufacturer has determined and recorded the correct 4-digit hexadecimal signature values for each point in the circuit. These are usually given in the equipment logic diagram. To troubleshoot the equipment, you connect the signature analyzer input to each point in the circuit, and a 4-digit LED display shows the signature obtained at each point. You can then compare the signature to the correct value on the logic diagram. If an incorrect signature is obtained, a fault is indicated.

Figure 11-27 shows a simplified block diagram of a signature analyzer. Note that the main element is a 16-bit shift register. The serial input data from the input point test is applied to the shift register through an exclusive OR gate. Outputs from the 7th, 9th, and 12th flip-flops in the shift register are fed back through exclusive OR gates and are exclusive ORed with the signal. After a specified number of inputs have been sampled, a fixed signature is generated.

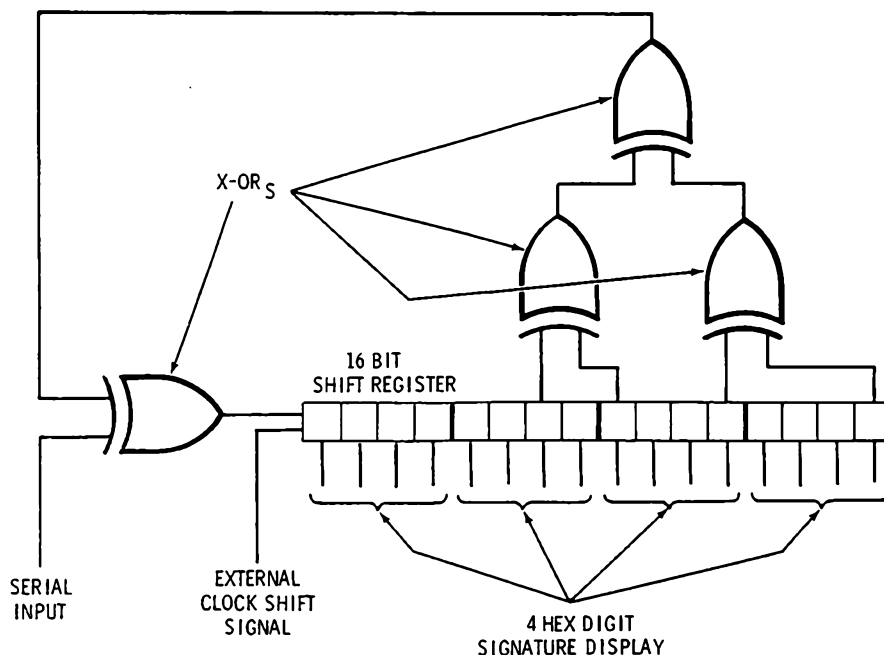


Figure 11-27
Basic circuit of a signature analyzer.

The 16-bit binary number stored in the shift register is the signature. That number is divided into four 4-bit segments to form a 4-digit hexadecimal number. Hexadecimal numbers, as you recall, feature 16 digits, the numbers 0 through 9 and the letters A through F. Typical signatures that may be displayed are 5C92, 70A4, B283, and F6D1.

The primary benefit of a signature analyzer is that it requires no operator interpretation. The signature is either correct or incorrect. For that reason, the signature analyzer is an ideal test instrument for unskilled technicians or production line workers to use in locating faults in digital equipment.

Self Test Review

16. Multimeters, logic monitors and logic probes are best for _____ testing.
17. DC voltages are best measured with a _____ multimeter.
18. Logic levels are best detected with a _____ multimeter.
19. For the most accurate logic level measurement, a multimeter with _____ is best.
20. To measure propagation delay, a _____ oscilloscope must be used.
21. Noise and distortion are best detected with a (an) _____.
22. The test instrument used to view all IC pin voltages simultaneously is called a _____.
23. Logic probes can make both static and dynamic tests.
 - A. True
 - B. False
24. The output response of a logic probe input is indicated on a _____.
25. Logic probes are typically battery powered.
 - A. True
 - B. False
26. An open circuit is usually indicated by _____ or _____ output lights on a logic probe.
27. Periodic pulse inputs are designated on a logic probe by _____ indicator lights.
28. The instrument used to generate a test pulse for a logic circuit is called a _____.

29. Shorts in digital circuits are easily detected by a _____ .
30. A logic probe detects _____ pulses; a current tracer detects _____ pulses.
31. The test instrument that samples, stores and displays multiple digital inputs simultaneously is called a _____ .
32. The three display modes of a logic analyzer are:
- A. _____
 - B. _____
 - C. _____
33. Most logic analyzers can display up to _____ waveforms simultaneously in the _____ mode.
34. Logic analyzers can “look at” up to _____ input signals at a time.
35. The display mode that uses D-to-A converters in a logic analyzer is the _____ mode.
36. Binary 0's and 1's are displayed in the _____ mode.
37. A signature analyzer has multiple data inputs.
- A. True
 - B. False
38. The output of a signature analyzer is a 4-digit _____ number.
39. The main circuit element in a signature analyzer is a _____ .
40. The signature output from a logic analyzer in the map mode and the signature output from a signature analyzer are the same.
- A. True
 - B. False

Answers

16. static
17. digital
18. analog
19. low voltage resolution
20. dual-beam
21. oscilloscope
22. logic monitor
23. A. True
24. lamp, light or LED
25. B — False. — Logic probes get their power from the circuit under test.
26. off or dim
27. flashing
28. logic pulser
29. current tracer
30. voltage, current
31. logic analyzer
32. A. Timing
B. Data
C. Map
33. 16, timing

- 34. 64
- 35. map
- 36. data
- 37. B — False. — A signature analyzer has one input.
- 38. hexadecimal
- 39. shift register
- 40. B — False.

PROCEDURES FOR DIGITAL TROUBLESHOOTING

The main objectives in troubleshooting are to repair the defective equipment as soon as possible at the lowest cost. This is particularly true of operating equipment, because the loss of that equipment keeps people from doing their jobs. The loss of a computer, a piece of test equipment, or some other system will often reduce productivity to zero. It is essential that the problem be located quickly and the equipment put back into operation.

The basic procedure for digital equipment troubleshooting involves three major steps. These are data collection, fault isolation, and repair. Let's consider each of these steps in more detail.

Data Collection

Data collection is the gathering together of all the information that you will need to service the equipment. Any information that you can locate with regard to the equipment before you make initial tests will usually simplify and speed up the troubleshooting process. Many individuals skip this step altogether and go directly to the equipment to make tests and repairs. This is OK if you are already familiar with the equipment and how to repair it. But if you are working on a piece of equipment for the first time, this information gathering step is extremely valuable.

One of the first things you want to locate is the equipment documentation. Documentation refers to all the operation and service manuals, logic and schematic diagrams, specification, operational procedures, and parts lists. Most digital equipment is supplied with manuals and information of this type. As complex as most digital equipment is, it is practically impossible to service without the manuals.

After you have collected all of the available documentation, go through it carefully. Review the manuals to familiarize yourself with what the equipment does and how it works. It is particularly important that you determine the function of the main operating controls. You don't want to introduce "cockpit" problems because of your lack of knowledge about the operation. You don't have to read the manuals all the way through, but you should familiarize yourself with their content. You will most likely need diagrams that show physical layout, logic diagrams, and schematics if you are going to do any detailed troubleshooting.

Often another source of data will be service and repair records. On some equipment, detailed logs are kept on repairs and general servicing. If the equipment you are working on has a service history, it will sometimes give you a clue as to the problem. In some equipment, the same things seem to fail over and over again. If this is the case, chances are, the same failure has occurred again. It will be easier for you to locate it.


If there are others who have serviced the equipment before you, you might ask them for their help. Perhaps they can give you some tips on how to approach the troubleshooting or a hint as to the possible problem. It is easier to service equipment for the first time with the assistance of someone who has already done it before. You will learn your way around the equipment faster and usually locate the problem a lot easier.

Finally, you will probably need IC, transistor, and other component specifications. The pin-out data on the TTL, CMOS, or ECL IC used in the equipment is essential to detailed troubleshooting. You may need specifications on transistors, diodes, capacitors, and other components. You can get all of this data from manufacturer's catalogs and data sheets. Sometimes this information is included with the equipment manuals.




Isolating the Problem

After you have familiarized yourself with the equipment through documentation, the next step is to approach the equipment and locate the problem. This is the most time-consuming and difficult part of all troubleshooting. In fact, it is the very heart of digital equipment servicing. The actual repair of the equipment is generally fast and simple. Usually the problem is found to be one simple component, perhaps costing only pennies.



Repair may involve unplugging an IC and plugging in a new one. Total repair time may only be minutes. But finding that problem can sometimes be difficult. The actual amount of time that it takes could vary from minutes to days. If you are working on a small piece of equipment with minimum complexity, you may be able to find the problem within minutes or several hours. Then again, if you are working on a large system, it may take days to isolate the trouble. In large systems, it is often necessary to narrow the problem down in stages. Most large digital systems are subdivided into major sections. These subsystems can be worked on individually once the problem has been isolated to one of them. In any case, there is no way to predict how long it is going to take to isolate the problem. It depends upon the nature of the problem, your own experience, and in many cases, choosing the best procedure for the particular problem.

There is no formal or established procedure for troubleshooting any kind of electronic equipment. The actual approach to troubleshooting will vary depending upon the equipment you are working on. In most cases, however, the troubleshooting procedure comes down to nothing more than a logical and sequential reasoning process. One way of looking at it is a cause-and-effect process. You notice the effect and attempt to determine what causes it.




There is no way to give you a foolproof step-by-step procedure to follow. However, in most digital troubleshooting, there are specific steps you can take in locating the problem. The following paragraphs list a sequence of nine procedures that you can follow in locating most problems. You may have to vary the sequence of them depending upon your needs, but in most cases, they should reveal the problem. These steps are outlined in Figure 11-28.

1. Operate the equipment.
2. Look for the obvious.
3. Run diagnostics.
4. Use your senses.
5. Check for power.
6. Test for clock.
7. Signal tracing.
8. Substitution.
9. Testing.

Figure 11-28
Steps in digital troubleshooting

OPERATE THE EQUIPMENT


The first step in the troubleshooting procedure is to turn on the equipment and attempt to operate it normally. Use the equipment as it would ordinarily be used and verify its operation. The knowledge you gained from the documentation should allow you to operate the equipment properly, and at the same time, determine if it is working. Check out the equipment thoroughly in all of its operating modes. What you are trying to do in this step is to verify the reported failure and duplicate or repeat the problem for your own observation.



Once you are able to duplicate the problem, you can observe for yourself what is or is not happening. Be sure to make detailed notes at this point on the symptoms, control settings, meter or display readings, or other information you can gather.

LOOK FOR THE OBVIOUS


During the first step, where you are operating the equipment and verifying the defect, you should be alert for clues to improper operation. This is a good time to look for the simple and obvious problems that so often are interpreted as major equipment failures. For example, the first thing you should look for is “cockpit” problems. Is the equipment being used properly? Are the controls set correctly? Are the readouts and displays being interpreted correctly? Go back to the manuals for answers to these questions if you have to. If someone else is operating the equipment, ask them to repeat the operation and to verify the failure themselves. Sometimes you may find that the operator simply isn’t using the equipment correctly or interpreting the result properly.



During this time, you should also look for things such as loose or broken cables. After an extended period of use, a cable or wire may break, or a connector may become loose or altogether disconnected. You may even find the classic problem of complete equipment failure. That is, the equipment may not operate simply because the power cord is not plugged into the outlet.

This is also a good time to disconnect any external equipment that may be attached to the equipment you are working on. Many pieces of test equipment or other instruments are used in conjunction with others. As a first step in isolating the problem, you should disconnect external units one at a time and note the result. Sometimes the external units have been incorrectly connected. You may also find that the external equipment has failed rather than the unit you are working on, which may cause the main unit to appear faulty.

At this point, the troubleshooting process can take many different courses. If the equipment simply isn’t operating at all, you would most likely suspect an AC or DC power failure. In this case, you would go directly there and check for power. On the other hand, if the equipment does seem to have power and is operating, although incorrectly, you may want to take another approach. In any case, you will need to vary the steps of the procedure to suit your particular situation.



RUN DIAGNOSTICS

A lot of digital equipment, particularly computers and large digital systems, have special diagnostics tests you can run to help locate the problem. In computers, the diagnostic tests are usually programs stored in a floppy disk or some other type of mass media. In any case, if you are dealing with a computer, running the diagnostics will usually locate the problem very quickly.

In other types of digital systems, the diagnostics programs are often built in. The diagnostic tests may be stored in a ROM. These diagnostics are self-testing procedures that you can usually initiate with a switch setting or a pushbutton. The whole idea of a diagnostic is to have large systems test themselves. Diagnostics are valuable in helping to isolate a problem quickly.

As digital equipment continues to get larger and more complex with advances in technology, more manufacturers are building in self-testing procedures and fault-isolation aids. Some equipment is even designed to contain small indicator lights that will turn on to indicate the specific point in a circuit where the equipment failed. In critical digital equipment such as that used by the military or in space operations, high reliability and rapid repair are absolutely essential. Use these built-in diagnostics; they will save you a lot of time. Usually the equipment documentation will spell out in detail what is being tested and how.

USE YOUR SENSES

At this point, you are probably beginning to narrow the problem down. You should have a clue as to what part of the equipment is defective. Now is the time to begin zeroing in on the fault.

By this time, you would probably begin to open the equipment; open cabinet doors, pull out the chassis on a workbench, or otherwise try to get to the actual circuitry. In any case, you want to be able to observe the equipment in operation.

Here, you should use your normal senses of sight, touch, smell, and hearing. For example, by carefully looking over the equipment, you may be able to spot the problem immediately; a broken wire, a loose connector, a printed circuit board that is not plugged in properly, or some other physical problem. You may see a cooling fan that is not operating or an excessively dirty air filter. Careful visual observation will often turn up a wiring error on a breadboard prototype like that shown in Figure 11-29. Take your time and look everything over carefully to be sure that it is as it should be.

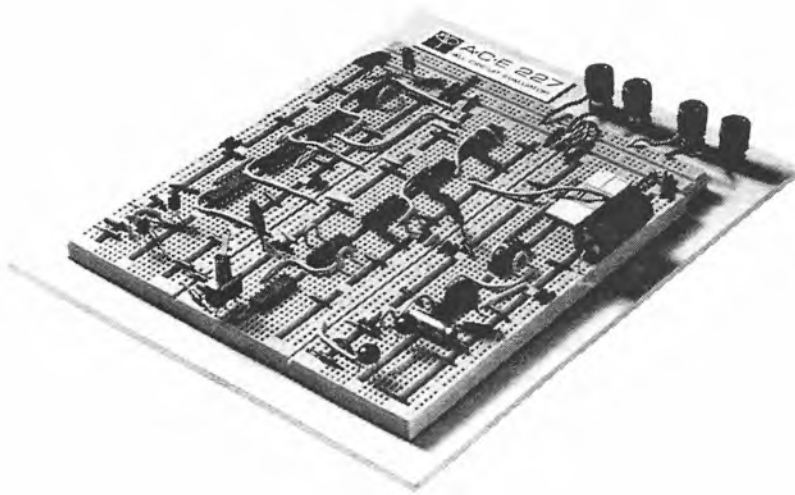


Figure 11-29
Typical breadboard prototype. Wiring errors are the
main problem.
Photo courtesy AP Products.

Smell is another sense that will often give you a clue to the problem, particularly in electronic equipment. One of the first things that you will notice is a burned smell that usually indicates a defective component. Burned resistors and transformers give off a unique odor. You may even be able to see smoke coming from the defective component or see an actual charred device.

The sense of touch is another help in locating problems. What you will be looking for by your touch is excessively hot components. You may want to gradually touch every component possible to see if it is excessively warm, but be careful not to burn yourself. Most digital ICs, for example, normally run warm to hot. Any component that is too hot to touch comfortably is usually defective.

Your sense of hearing, while not the most important in troubleshooting, will sometimes help give away the problem. For example, you may not hear the cooling fan that is supposed to be running. You may also hear crackling sounds that may indicate a burning component. Rattling sounds could indicate disconnected cables or similar problems.

Observation is a simple procedure that will many times lead you directly to the problem. It is often overlooked as part of the troubleshooting process. Take your time to do it, as it could eliminate a lot of work.

CHECK FOR POWER

At some point in the troubleshooting process, you may want to test to see that you have the proper power to the equipment. When the equipment is not operating at all, the usual reason is that proper power is not being supplied. This could be as simple as the AC line cord not being plugged in, to a major failure in the power supply. So, before you begin detailed testing procedures of the logic circuitry itself, be sure that proper AC and DC power exists.

Using a standard analog or digital multimeter, you should first check to see if the AC power is being supplied to the unit. Make sure the line cord is plugged in and that AC power is reaching the input to the power supplies. The most common reason that the power supplies are not getting AC power is a blown fuse. Typically a fuse is connected in the AC power line. If a fault or overload occurs, the fuse blows to protect the equipment, and often, this is the only problem with the equipment. When you replace a fuse, make sure you use one of the same value.

Some types of equipment use circuit breakers rather than fuses. Like the fuses, the circuit breakers are usually located on the back of the equipment. The circuit breaker is either a switch or pushbutton that you can reset which will usually put the equipment back in operation quickly. Sometimes, there is no actual failure; a transient overload will simply blow the fuse or trip the breaker, shutting off the equipment.

If replacing the fuse or resetting the breaker causes the fuse to blow again, or the breaker to trip, then a serious problem is indicated. It usually means that there is a fault in the main power supply.

Using a multimeter, next check to see that all of the DC voltages coming out of the power supply are correct. Most digital systems have at least one power supply that is usually a 5-volt supply. Many systems have more than one power supply. Typical supply voltages are +5 volts, + and -12 volts, + and -15 volts and 24 volts. Using the schematic diagram as a guide, make sure the power supply voltages are within tolerance. If the supply voltages are not there, then you know you have a power supply problem. At this point, you should remove the power supply and repair it. On the other hand, if all of the supply voltages are what they should be, then you know that the problem lies elsewhere. However, if the supply voltages are lower than they should be, it is probably because of excessive current loading by one or more circuits. Here, you should disconnect one circuit at a time until the supply voltage returns to normal. In this way you isolate the defective circuit.

A typical digital power supply is shown in Figure 11-30. Power supply failures are common in digital equipment.

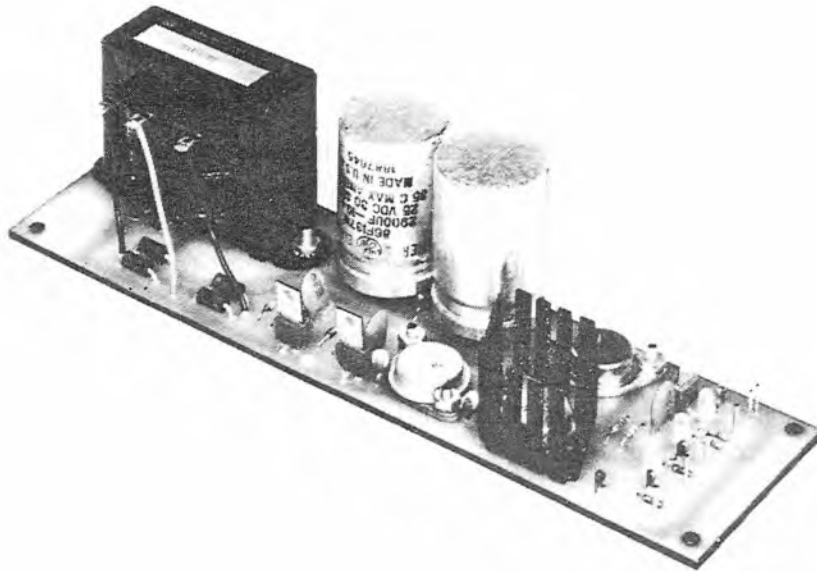


Figure 11-30

Typical digital power supply. Power supply failures are common in digital equipment.

TEST FOR CLOCK

Virtually all digital equipment uses a clock or master oscillator to operate the circuits. This can be a simple astable multivibrator or a temperature-controlled quartz crystal oscillator. In any case, the equipment will not operate at all if it has no clock signal. Once you have verified that all the proper supply voltages are there, checking for the existence of a clock signal is the next step.

You can check for the existence of a clock quickly and simply with a logic probe. If you get a favorable indication, chances are the clock is operating properly. However in many cases, it is best to make a more thorough check; use an oscilloscope to verify the integrity of the clock signal. By looking at the clock signal on the oscilloscope, you can measure its frequency, pulse width, voltage level, and other characteristics to see if they are correct. In most digital equipment, the amplitude, pulse width, and frequency of the clock signal is critical. Even minor variations can cause problems.

If no clock signal exists or the clock signal characteristics are incorrect, then you have at least partially isolated the problem. You know the trouble is in the clock circuit. On the other hand, if the clock is operating properly, the trouble lies in another part of the equipment.

SIGNAL TRACING

At this point, the troubleshooting can again take many paths. Depending upon what the equipment is and how it operates, there are all kinds of procedures that you may wish to try. One of the most common and useful, however, is signal tracing. Signal tracing is the process of tracking a signal through the equipment from input to output. Most digital equipment is designed to accept input signals, process them in some way, and generate useful output signals. Signal tracing is the process of simulating input signals and tracing them through the circuit to see that they produce the desired output result.

Signal tracing requires some form of input signals. If the equipment itself is receiving its input from a keyboard, transducers, or other sources, and they can be used in the troubleshooting process, by all means use them. However, in some cases you may not be able to duplicate the input signals exactly. In this case, you will need to simulate them with a test instrument such as a logic pulser or signal generator. Binary word generators are also available to produce artificial input data for equipment testing.

To verify that appropriate outputs are occurring, you can often use the readouts and displays built into the equipment, if they exist. In many cases, there are LEDs, 7-segment readouts, or CRT displays that can be observed. If not, you may need a multimeter, oscilloscope, logic probe, or logic analyzer to verify that you are getting the correct results.


As you generate input signals, you will typically begin tracing them through the circuitry if you are not getting the correct output. The intent is to follow the signal as far through the circuitry as you can. The point in the circuit where you lose the signal is where the trouble exists. At this time, you may not know exactly why the signal stops, but at least, you have generally isolated the problem and you can begin more detailed testing. At this time, logic diagrams, logic clips, logic probes, etc. can be the most valuable.



SUBSTITUTION


Substitution is one of the fastest and easiest methods of finding the problem in digital equipment. Once you have isolated the problem to a general area such as a printed circuit board or integrated circuit, you can often repair it quickly by a simple substitution process. Let's say that you have narrowed the problem down to a particular printed circuit board in the equipment. If a new replacement board is available, substitute it for the old one. If this solves the problem, then you know the difficulty lies on that board. The substitute board puts the equipment back in operation quickly. You can always repair the defective board later. In many large systems, spare printed circuit boards are made available for just such situations.

Substitution also works at the component level. For example, you may isolate the problem to one or two integrated circuits. If these integrated circuits are installed in sockets, you can quickly replace them with new ones. If the ICs are not in sockets, you will have to unsolder them. This is time-consuming and potentially damaging, but it may have to be done. In any case, substituting an IC is a fast way to verify if it is defective.



You can also replace other components such as defective transistors, diodes, capacitors, or resistors. By making circuit measurements, you can often identify the defective component. For example, if the voltages at the emitter, base, and collector of a transistor are not correct, you may suspect a defective unit. This is the classical way of troubleshooting. That is, you make careful circuit tests and measurements, and attempt to pinpoint the defective component. However, this often takes more time than simply substituting a component that you suspect of being defective.

Substitution is, in many ways, a "brute force" troubleshooting technique. It is often a little more expensive, since substitute components are needed. On the other hand, there is probably no faster way to get a piece of equipment operating.



TESTING

If substitution does not solve the problem, the next step is detailed logic testing. At this point in the procedure, you have isolated the problem to a particular subsystem or circuit. The problem will be revealed by more detailed testing.

The two basic types of digital testing are static and dynamic. In static logic circuits, the clock is usually disabled. At this time, all of the logic levels in the circuit are stable. With this condition, you can look at them with a multimeter or a logic probe.

Many systems allow you to disable the clock signal so that static testing is possible. Other systems provide a manual pushbutton which you can use to trigger the circuitry one pulse at a time. This pushbutton usually is substituted for the clock so that you can sequence through the logic operation a step at a time by pressing the button. Such single-step operation is extremely helpful in troubleshooting.

Dynamic testing means testing the equipment while it is operating under normal clocked conditions. The system clock runs at its normal rate. You will then need to use an oscilloscope or logic analyzer to isolate the problem. A logic probe may be helpful in some cases, but for most dynamic testing, it does not give the kind of detail usually required.

Dynamic testing is the preferred approach, although it is the most difficult. Sometimes a logic circuit operates properly when it is static or in a slow-speed, single-stepped mode. However, the equipment may not operate properly when it is run under normal high-speed conditions. Dynamic testing is preferred because it shows up problems such as noise or propagation delay troubles that create timing problems.




Making the Repair

Perhaps the easiest part of servicing digital equipment is making the actual repair. You will probably spend most of your troubleshooting time just isolating the problem. Once you have found it, the repair may take only a few seconds, or in many cases, not more than minutes. For example, you may fix the equipment by replacing a PC board. Perhaps the problem is to resolder a broken wire or make a simple timing adjustment in the clock circuit.


The most common repair, however, is replacing a defective component. This usually amounts to changing an IC, replacing a transistor, or installing a new capacitor. Such repairs are generally easy, but there are two important points to consider in making the repair. First, you should take care if you do any unsoldering and soldering. Second, if you must deal with MOS integrated circuits, proper handling is essential.

UNSOLDERING AND SOLDERING



Many times making the repair means removing a damaged component from a printed circuit board. Unsoldering a component may be difficult, and it can also cause a lot of damage if done improperly.

The best way to unsolder a component is to use one of the special solder removing tools commonly available. Usually a vacuum bulb is used to pull the solder off the connection as you heat it.



Another approach is to use solder wick. Solder wick is a special type of braided wire that is coated with flux, a rosin that will quickly attract solder. Solder wick is an ideal way to remove solder from a connection as it is heated. See Figure 11-31.

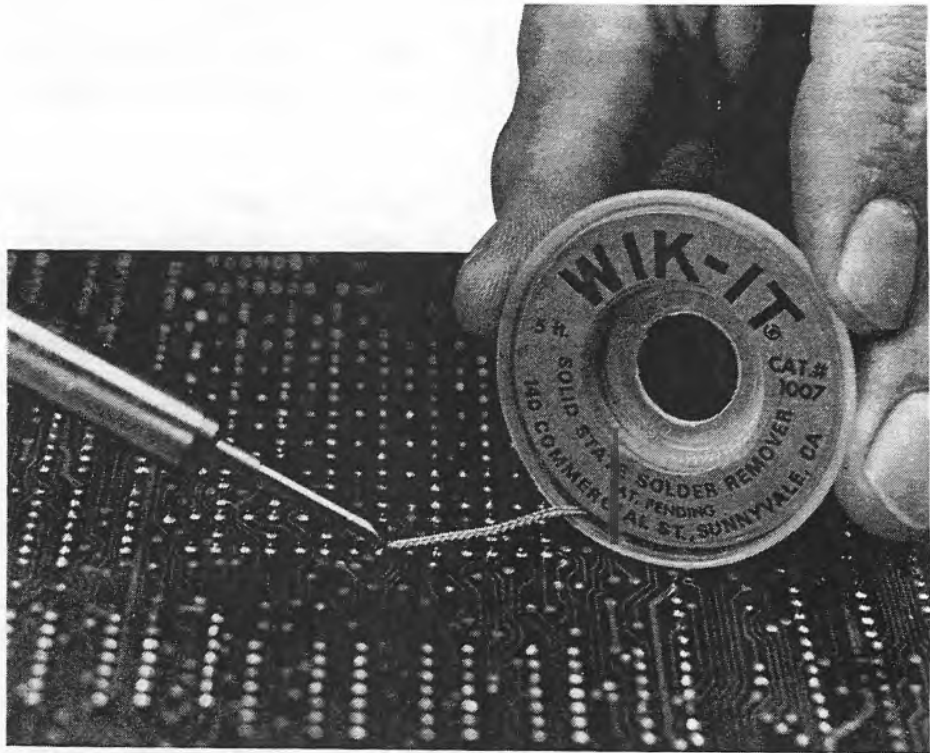


Figure 11-31

Solder wick removes solder fast and completely from PC boards making IC's and other components easy to replace.

Special unsoldering tools are also available, such as special tips on soldering irons that allow all pins on an integrated circuit to be heated and unsoldered simultaneously. While such tools are useful, it is usually best to remove solder from connections before you attempt to take a component off the circuit board. In any case, be extremely careful and avoid overheating the circuit board. Excessive heat will cause copper lands to peel off the board and cause further damage.

When you take a component such as a transistor, diode or integrated circuit from a PC board, be sure to note the correct orientation of the pins. In fact, you may want to write this down before you forget; later, you can install the replacement correctly. It is frustrating to install the new component incorrectly and find that the equipment still doesn't work.

When you replace the component, resolder it carefully. Be sure there is sufficient solder on each pin, but be careful not to overheat the connection and cause damage. Further, watch for excessive solder that could cause solder bridges (shorts between adjacent pins caused by too much solder or sloppy soldering techniques).

MOS DEVICES

Today almost all digital equipment uses some type of MOS integrated circuit. These may be P-channel or N-channel devices or CMOS circuits. All of these devices can be damaged by static electricity. When you replace an MOS device, you must handle it properly to avoid damaging it.

Most new MOS devices are supplied by the manufacturer packed in a conductive foam. The IC pins are usually pushed into the foam, which essentially keeps all of the pins shorted together so that static electricity does not get to them.

When you are ready to install the new device, carefully remove the IC from the foam while touching the PC board in which the new device will be installed. Touching the IC in the foam and the PC board simultaneously, prevents static electricity arcing. You can usually prevent damage to MOS circuits due to static electricity by grounding the equipment, the tools, and yourself. Be sure that the equipment you are working on is properly grounded. You may also want to ground the soldering iron with a clip lead. Many assembly line workers ground themselves with a wrist strap. If you take such precautions, you should be able to reinstall the IC without damaging it.

TESTING

After you have made the repair, you will need to reassemble the equipment. Then you will make the final tests to see that it is operating properly. You will probably want to run through the same procedures you did initially. Try to repeat the fault if you can. Again, verify that the equipment operates exactly as it should in all possible modes. You may even want to run the diagnostic tests again if available. Better still, let the normal user of the equipment try it out again to be sure that it functions properly. As you have verified its operation, you can close up the equipment and reinstall it.

Self Test Review

41. All of the manuals and printed data available on a piece of digital equipment is called _____.
42. Troubleshooting is a _____ and _____ reasoning process.
43. You can often locate the problem by simple observation using your _____.
44. Built-in tests that automatically help locate problems are called _____.
45. A blown fuse or tripped breaker can keep AC power from reaching the _____.
46. The fastest method of troubleshooting is _____.
47. A logic pulser can be used as the input, and a logic probe as the output, to perform a common testing method called _____.
48. Simulating the clock with a pushbutton in static testing is called _____.
49. Unsoldering is best done with _____.
50. _____ ICs can be damaged by static electricity if not handled properly.

Answers

- 41. documentation
- 42. logical, sequential
- 43. senses
- 44. diagnostics
- 45. power supplies
- 46. substitution
- 47. signal tracing
- 48. single stepping
- 49. solder wick
- 50. MOS

EXPERIMENT 26

Practical Digital Troubleshooting

OBJECTIVE: *To demonstrate the operation of a logic probe and to show its use in troubleshooting practical digital circuits.*

Materials required:

Heathkit Digital Design Experimenter
Analog or Digital Multimeter
Heathkit IT-7410 Logic Probe or equivalent (optional)
1—14495-1 LED Driver IC (443-1802)
1—74LS00 TTL IC (443-728)
1—74LS42 TTL IC (443-807)
1—74LS193 TTL IC (443-815)
1—7-Segment LED (411-885)

Procedure

In the first part of this experiment, you will become familiar with a logic probe. The experiment is designed specifically to show the features of the Heath IT-7410 Logic Probe, although almost any logic probe can be used successfully. If you do not have a logic probe, simply skip the first part of this experiment and go on to the second part of the procedure beginning at Step 11. Logic probes are one of the most often used pieces of digital equipment, and it is essential to become familiar with them. We encourage you to purchase a logic probe or borrow one for this experiment.

1. Connect your logic probe to the ET-3200 Experimenter with short pieces of hookup wire. Connect one end of the hookup wire into the ground and +5 volt sockets on the Experimenter and the other ends to the power and ground clips of the logic probe.
2. Apply power to the ET-3200 Experimenter; then touch the tip of the logic probe to the ground connection on the Experimenter and observe the result. Next, touch the logic probe to the +5-volt supply voltage. Note the probe indications below.

Ground _____.

+5 volts _____.

3. Touch the logic probe to the normal output switch A and note the result. Depress logic switch A and again note the result.

Switch normal _____.

Switch depressed _____.

4. Press the reset button on the logic probe and be sure the memory light is off. Touch the probe tip to the normal output of logic switch A. Press and release the switch once, noting the logic switch indications, including the state of the memory lamp.

Memory light state _____.

5. Press the reset button of the logic probe, turning off the memory light. Touch the logic probe to the complement output of logic switch A. Again press the switch and release it, noting the logic probe indicator and memory light changes, if any.

Memory light state _____.

6. Set the logic clock to the 1 Hz position. Touch the tip of the logic probe to the clock output. Note the response on the logic probe. Count the number of pulses occurring to determine the frequency of oscillation of the probe lights. One way to do this is to count the number of times the pulse light goes on in a 10-second period, then divide this number by 10 to get the frequency in Hz.

Frequency _____ Hz.

7. Set the clock switch to the 1kHz position. Touch the probe to the clock output and again note the frequency of operation by counting the output pulses over a fixed time interval. Repeat for the 100 kHz clock position. Explain your results.

Frequency (1 kHz position) _____ Hz.

Frequency (100 kHz position) _____ Hz.

8. Wire the circuit shown in Figure 11-32. Connect the power (14) and ground (7) pins to + 5 volts and ground on the Experimenter.
9. Touch the logic probe to pins 1 and 2 of the 7400 IC and record the logic probe indication. Explain the reason for this indication.

Pin 1 _____.

Pin 2 _____.

10. Touch the probe output to pins 3 and 6 and note the probe output indication.

Pin 3 _____.

Pin 6 _____.

Discussion

In this part of the experiment, you demonstrated how the logic probe can be used to detect various logic levels and digital signals. You demonstrated this first by touching the probe to the ground and +5-volt power supply outputs on the Experimenter. Touching the logic probe to ground causes the white light indicator to turn on, thus indicating a binary 0. Touching the +5-volt supply line turns on the red probe light, thereby indicating a binary 1 level.

Next, you measured the output of logic switch A with the logic probe. The normal output is usually a binary 0 before you press the switch. When you press the switch, the normal output goes to a binary 1. Releasing the switch causes the output to go to a binary 0. Of course, the complement output gives the opposite results. The output is a binary 1 before you press the switch, and changes to binary 0 when the switch is pressed.

The memory indicator on the logic probe is designed to “remember” when a logic level transition occurs, so you must manually reset the memory light before you use it. If the point to which the logic probe is touched switches from binary 0 to binary 1, or from binary 1 to binary 0, the memory light will turn on. You should have noted that the memory light did go on when you pressed logic switch A. The light should have turned on for both cases of 0-to-1 and 1-to-0 logic level changes.

Next, you used the logic probe to indicate the presence of a clock signal. With the Experimenter clock in the 1 Hz position, the logic probe indicator lights followed the clock output changes. The clock switches at approximately 1 Hz or 1 output pulse per second.

With the Experimenter clock in the 1 kHz or 100 kHz position, the logic probe flashed at approximately a 3 to 5 Hz rate. This flashing rate is set by the internal oscillator of the logic probe. Any frequency higher than 10 Hz will automatically be detected and indicated as a 5 Hz flashing light on the logic probe.

Finally, you used the logic probe to measure typical input and output logic levels on a standard TTL circuit. First, you measured the open inputs of a 7400 TTL gate. Since the two inputs are not connected to binary 0 or binary 1, you will actually be measuring the internal voltage level of the TTL input circuit. This will usually be about +1.5 volts. The probe will simply detect this as an invalid logic level or an open circuit. On the IT-7410 Logic Probe, this is indicated by **both** binary 0 and binary 1 lights being off. On some logic probes, the indicator lights will glow dimly at a brilliance level somewhere between the binary 0 and binary 1 levels.

On a TTL NAND gate, open inputs are interpreted as being binary 1 levels. Therefore, the output of gate 1 at pin 3 should have been at a binary 0 level. This in turn is connected to gate 2 in Figure 11-32. This gate acts as an inverter, generating an output at pin 6, which is a binary 1 level. The logic probe should have indicated this correctly.

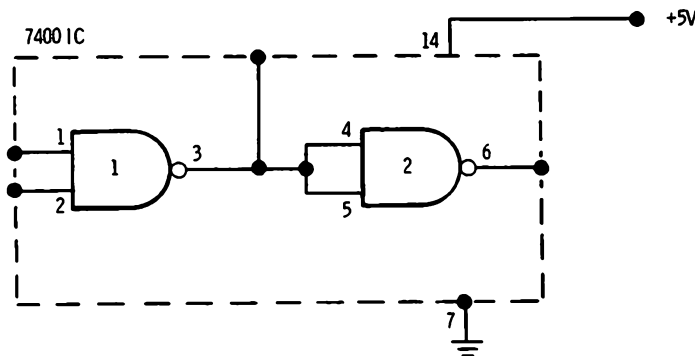


Figure 11-32
Diagram of circuit for experiment 26, step 8.

Procedure (continued)

In this part of the experiment, you will wire a typical digital circuit. You will determine how that circuit works and then perform a variety of troubleshooting procedures on it. The idea is to show you a procedure or step-by-step method you can use to test a circuit and verify its operation. If the circuit does not work, you will perform tests to isolate the problem. Then you will repair the circuit and verify that it is operating correctly.

11. Construct the circuit shown in Figure 11-33. Take your time to build the circuit exactly as shown. You should know in advance that the circuitry wiring has several "errors" in it. We hope that you will not spot the errors, but if you do, go ahead and make the wiring exactly as shown.
12. Observe the circuit in Figure 11-33. Analyze the operation of the circuit to determine how it works. Find out the specific function of the circuit before going on. As a starting point, assume logic switch A, the reset button, is pressed, then logic switch B, the start pushbutton, is pressed. Analyze the operation of the circuit after those initial inputs. The clock (CLK) input to gate 1 can be any frequency.
13. Set the ET-3200 Experimenter clock to 1 Hz. Apply power to the circuit and test its operation. Make whatever measurements you need with a multimeter or logic probe to satisfy yourself that the circuit is operating properly. If the circuit does not function properly, explain or list the symptoms you observe.
14. Begin the troubleshooting procedure, using a multimeter or logic probe as you require. If you do not have a logic probe, you can simply use one of the logic indicator LEDs on the Experimenter. Often, nothing more than a simple LED indicator like this is needed for simple digital troubleshooting. In this step, specifically check for the existence of power supply voltages and the clock signal.

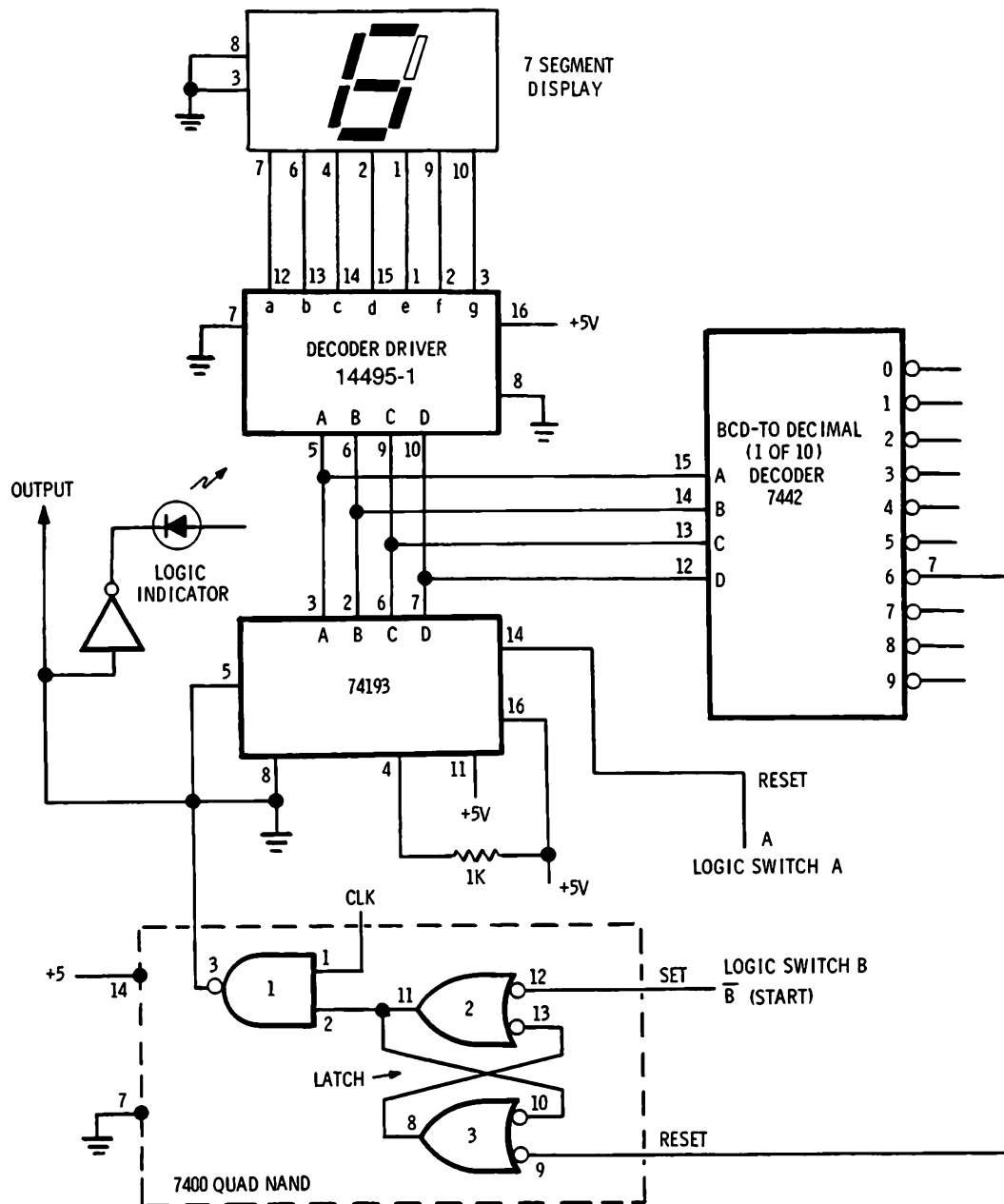


Figure 11-33
Circuit for step 11.

15. Perform whatever troubleshooting steps you can think of to help locate the problem. As a hint, use signal tracing techniques where the clock is assumed to be the input.
16. After you have discovered the problems with the circuit, make the repairs. In this case, you will be able to repair the circuit by correcting the wiring. After you repair the circuit, check the operation of the circuit to be sure that it performs properly.

Discussion

The first part of the experiment is to determine the function of the circuit in Figure 11-33. It should be obvious to you at this time, that it is almost impossible to intelligently troubleshoot a digital circuit without knowing how it operates. Earlier, we made the point that it is absolutely essential that you understand the operation of the circuit to avoid “cockpit” errors. Further, you simply do not know whether it is operating properly if you do not know what it does. In many cases, the equipment documentation will give you all the information that you need. In other cases, such as this one, no information will be given. You will simply have to determine the operation of the circuit yourself by analyzing the logic diagram.

The circuit in Figure 11-33 is a pulse burst generator. That is, when you press switch B, the circuit generates a fixed number of output pulses and then halts. The number of pulses in the burst generated is determined by which output on the 74LS42 1-of-10 decoder is selected.

The circuit operates when logic switch A is pressed. This causes the 74LS193 binary counter to be properly reset prior to operating. The latch made up of gates 2 and 3 is reset. The output of gate 2 is low, thereby inhibiting gate 1. Therefore, the clock signal does not pass through gate 1 to the counter. When the start button is pressed, the latch is set. This enables gate 1 and clock pulses pass through it to the 74LS193 counter and the output. Each clock pulse increments the counter. The 14495-1 decoder driver observes the counter output and displays the counter’s content on the 7-segment LED.

The 7442 1-of-10 decoder monitors the counter outputs. As soon as the selected number of outputs occur, the appropriate gate on the 7442 will go low and cause the latch to be reset. This causes gate 1 to be inhibited and the clock pulses to be stopped. The number 6 output of the 7442 1-of-10 decoder (pin 7) is used here; therefore, the circuit should generate six output pulses before stopping. You can observe the output pulses at the output of gate 1.

At the end of the burst, the 7-segment LED indicator shows the number of pulses generated. The cycle will repeat if you press logic switch A to reset the circuit and logic switch B to start it. You can verify the operation of the circuit for different numbers of output pulses by changing the connection on the output of the 7442 decoder.

As you probably noted, the circuit does not function properly.

In step 14, you should have discovered that there were no power and ground connections on the 7442 IC. In checking for power and ground, you should do more than just check for the presence of the supply voltage at the power supply output. You should verify that every IC in the circuit has power and ground applied to it. Usually the power and ground pins are noted on the logic diagram. You can very quickly and easily check the power and ground on each IC with a logic probe. Once you discovered that the 7442 did not have power and ground on it, you should have added these connections.

Again, you should have tested to see whether the circuit was operating properly; you probably discovered that it did not. The main symptom you should have noted was that once the circuit was reset with logic switch A and started with logic switch B, the state of the counter never changed. This was indicated by the LED display continuing to show zero even after the circuit was started.

At this point, you should have tried signal tracing. With the logic probe, you should have found that the clock signal was present at the input to gate 1. Measuring the output of gate 1 should have shown you that no clock pulse existed. This is the main circuit output and should have been one of the first places for you to check. Recall that most circuits have an input and an output, with the circuitry in between doing the necessary processing. By tracing the circuit, you should have found that we deliberately grounded the output of gate 1. By removing this wire, clock pulses will now reach the input of the 74193 counter.

At this point, the circuit should have operated properly.

UNIT EXAMINATION

The purpose of this exam is to help you review the key facts in this unit. The problems are designed to test your retention and understanding by making you apply what you have learned. This exam is not so much a test as it is another learning method. Be fair to yourself and work every problem first before you check the answers.




1. The most critical element in repairing digital equipment to prevent loss of productivity is:
 - A. Cost.
 - B. Proper test equipment.
 - C. Experienced repair personnel.
 - D. Fast service.
2. Which of the following components is most likely to fail first?
 - A. Switch.
 - B. Connector.
 - C. Fuse.
 - D. Diode.
3. Which of the following would cause a timing problem?
 - A. Propagation delay.
 - B. Power supply voltage.
 - C. Clock frequency.
 - D. Heat.

4. Which of the following is NOT an environment problem?
 - A. Power line spikes.
 - B. Heat.
 - C. Dirt.
 - D. Vibration.

5. Which of the following give the false impression of a hardware problem?
 - A. Mechanical failures.
 - B. Operator problems.
 - C. Excessive heat.
 - D. Software problems.

6. Which of the following is NOT an example of an internal IC failure?
 - A. Solder bridge.
 - B. Open weld.
 - C. Defective chip.
 - D. Shorted pins.

7. Short circuits can be caused by:
 - A. Broken pins.
 - B. Broken welds.
 - C. Defective IC socket.
 - D. Solder bridges.

- 
8. A scratched PC board can cause a/an:
- A. Short.
 - B. Open circuit.
9. Chemicals often cause:
- A. Shorts.
 - B. Open circuits.
10. Which of the following instruments are best for static testing?
- A. Oscilloscope.
 - B. Multimeter.
 - C. Logic probe.
 - D. Signature analyzer.
- 
11. Which of the following is NOT a dynamic test instrument?
- A. Logic monitor.
 - B. Oscilloscope.
 - C. Logic probe.
 - D. Logic analyzer.
12. An oscilloscope is best for measuring which of the following?
- A. Logic levels.
 - B. Noise.
 - C. Frequency.
 - D. Distortion.
- 

13. The test instrument that gives a visual output indication to a single logic input is called a:
- A. Logic monitor.
 - B. Logic pulser.
 - C. Logic probe.
 - D. Logic clip.
14. The circuit in Figure 11-34 can be used as a simple TTL logic probe.
- A. True.
 - B. False.



Figure 11-34
Circuit for Exam question 14.

15. A flashing output on a logic probe usually indicates which of the following inputs?
- A. Open circuit.
 - B. Binary 0.
 - C. Binary 1.
 - D. Pulse train.

16. The instrument used to generate an input stimulus to test a digital IC is called a:
- A. Logic pulser.
 - B. Current tracer.
 - C. Logic probe.
 - D. Signature analyzer.
17. Defects such as shorts in wired ORs and bus systems are easily found with a:
- A. Logic probe.
 - B. Current tracer.
 - C. Logic pulser.
 - D. Multimeter.
18. The test instrument that collects and stores many simultaneous digital inputs is called a/an:
- A. Oscilloscope.
 - B. Logic probe.
 - C. Logic analyzer.
 - D. Signature analyzer.
19. When the CRT display shows binary 0's and 1's, octal, or hex characters, which logic analyzer display mode is selected?
- A. Timing.
 - B. Map.
 - C. Graphic.
 - D. Data.

20. Which logic analyzer mode is best for recognizing specific data words or bit patterns?
- A. Timing.
 - B. Data.
 - C. Map.
21. The map mode in a logic analyzer generates which of the following outputs?
- A. 1's and 0's.
 - B. Timing waveforms.
 - C. Hex digits.
 - D. Signature pattern.
22. The dynamic test instrument with a single serial input and a 4-digit hex output display is called a:
- A. Logic analyzer.
 - B. Signature analyzer.
 - C. Logic probe.
 - D. Logic monitor.
23. The main circuit in a signature analyzer is a:
- A. RAM.
 - B. Sampler.
 - C. Multiplexer.
 - D. Shift register.

24. A signature analyzer is connected to a point in a circuit whose signature is 2E7A. The signature analyzer display shows 2E8A. The circuit is defective.
- A. True.
 - B. False.
25. The main causes of AC power not reaching the power supplies are:
- A. Tripped breaker.
 - B. Defective regulator.
 - C. Shorted power transformer.
 - D. Blown fuse.
26. The logic circuitry in a system is static. The problem could be a defective:
- A. Power supply.
 - B. Clock.
 - C. Output display.
 - D. Connector.
27. Tracking a logic pulse from input to output is called:
- A. Substitution.
 - B. Single stepping.
 - C. Signal tracing.
 - D. Current tracing.

28. One of the fastest ways to get defective equipment operating is:
- A. Dynamic testing.
 - B. Signal tracing.
 - C. Static testing.
 - D. Substitution.
29. The most common defect in breadboard prototypes is:
- A. No DC power.
 - B. Defective clock.
 - C. Wiring error.
 - D. Bad IC.
30. Touching the IT-7410 Logic Probe to an IC input pin causes both indicator lights to go off. The problem is a/an:
- A. Short.
 - B. Open circuit.
 - C. Noise.
 - D. Blown fuse.

EXAMINATION ANSWERS

1. D—Fast service.
2. C—Fuse.
3. A—Propagation delay.
C—Clock frequency.
4. A—Power line spikes.
5. B—Operator problems.
D—Software problems.
6. A—Solder bridge.
7. D—Solder bridges.
8. B—Open circuit.
9. A—Shorts.
10. B—Multimeter.
C—Logic probe.
11. A—Logic monitor.
12. B—Noise.
D—Distortion.
13. C—Logic probe.
14. A—True. A binary 1 will turn the LED on. A binary 0 will not light the LED.
15. D—Pulse train.
16. A—Logic pulser.

17. B — Current tracer.
18. C — Logic analyzer.
19. D — Data.
20. B — Data.
21. D — Signature pattern.
22. B — Signature analyzer.
23. D — Shift register.
24. A — True.
25. A — Tripped breaker.
D — Blown fuse.
26. B — Clock.
27. C — Signal tracing.
28. D — Substitution.
29. C — Wiring error.
30. B — Open circuit.